NCR

NCR I-8140

Manual de Instalación

NCR Intellectual Properties IP

I N D I C E

GENERALIDADES IDPS/SF, IDPS/SFE

- SISTEMA OPERATIVO IDPS/SFE
- RUTINAS DEL SISTEMA
- HERRAMIENTAS DE SOFTWARE

LENGUAJES DE CONTROL DEL SISTEMA

NORMAS PARA LA ASIGNACION DE 4501/4

RUTINAS DE UTILIDAD DEL SISTEMA OPERATIVO

- COMPILADOR COBOL
- DEBUG
- EDIT
- FLEXIT
- OVCOPY
- SELECT
- SORT
- SPECSORT
- SUMLG

TIPOS DE DISCO FLEXIBLE

ARCHIVOS EN DISCO FLEXIBLE

DIFERENCIAS COBOL IMOS/COBOL DPS

N C R I - 8140

SISTEMA OPERATIVO IDPS/SFE

DESCRIPCION GENERAL

El equipo NCR I-8140 dispone del sistema operativo IDPS/SFE (Interactive Direct Processing System/Small File Extended.

El sistema NCR I-8140 puede presentarse con una memoria inferior a 96 Kbytes, en cuyo caso será necesario utilizar el sistema operativo IDPS/SF (Interactive Direct Procesing System/Small File).

Las características de este sistema operativo son iguales a las del sistema operativo IDPS/SFE. La diferencia entre uno y otro sistema estriba en que el IDPS/SFE proporciona un tamaño de memoria de usuario no dependiente de las opciones hardware y software incluidas durante la generación del sistema, y en general este tamaño es superior al proporcionado por el IDPS/SF.

La conclusión inmediata es que, siempre que el sistema disponga de una memoria superior o igual a 96 Kbytes es recomendable utilizar el IDPS/SFE, mientras que se utilizará el IDPS/SF sólo en los casos en que esta memoria sea inferior a 96 KB.

Este sistema operativo, residente en disco flexible, proporciona todos los servicios necesarios para ejecutar tareas en monoprogramación en el hardware de este equipo.

El sistema permite la ejecución de programas de aplicación codificados en COBOL 74 o en BASIC, rutinas de utilidad, comandos del sistema (JCL) y mensajes de error, así mismo proporciona soporte para la ejecución de tareas internas, tales como operaciones de entrada, de salida, archivo de errores y carga de programas de aplicación.

Los programas de aplicación pueden estar preparados para ser ejecutados en modo interactivo o en "batch". El modo interactivo permite al programa obtener datos del operador, solicitándolos por pantalla; en batch, por el contrario, los datos que deben haber sido preparados con anterioridad, son procesados sin requerir acción alguna por parte del operador.

La unidad de trabajo ejecutada por el sistema ~~en~~ es el "job". Un "job" viene definido por una serie de comandos del sistema, también denominados instrucciones JCL que permiten al usuario especificar el programa que ha de ejecutarse, así como los archivos que serán utilizados durante la ejecución. Estos comandos pueden haber sido almacenados previamente en un archivo de disco flexible y ejecutarse mediante una sola instrucción.

El sistema operativo está compuesto por una serie de módulos independientes, que se combinan de acuerdo con la configuración indicada al generar el sistema.

RUTINAS DEL SISTEMA

Las rutinas que se describen a continuación facilitan la labor del usuario del sistema, resolviendo muy diversas situaciones sin requerir programación adicional.

CHANGE

Cambia el nombre físico de un archivo en el directorio de un disco flexible por otro proporcionado por el operador.

DELETE

Elimina lógicamente uno o todos los archivos de un disco flexible. Los archivos del sistema no pueden ser eliminados.

DISPLAY

Presenta en pantalla o impresora el contenido del directorio de archivos en disco flexible o etiquetas de cassette.

DUMP

Presenta en pantalla o impresora el contenido de un archivo o de un área especificada de un disco flexible o cassette en formato hexadecimal, ASCII o hexadecimal y ASCII.

FLEXIT

Inicializa un disco flexible o cambia una cabecera de volumen. La función de inicialización asigna una única dirección de sector y pista a cada uno de los sectores de un disco flexible en densidad sencilla o doble o cuádruple capacidad.

MOVE

Copia físicamente un archivo o todos los archivos de un disco flexible o cassette a otro. También imprime parte o todo un archivo. Esta rutina no copia en el archivo de destino los registros eliminados en el archivo de origen.

OVCOPY

Esta rutina copia los overlays del sistema operativo desde el disco Maestro a discos de datos o los overlays del sistema operativo más el sistema operativo residente desde el disco Maestro a discos de aplicaciones.

QUICK COPY

Copia el contenido completo (excepto los overlays del sistema operativo y el archivo de errores) de un disco flexible a otro disco flexible. Tanto el disco flexible de origen como el de destino deben ser de la misma densidad de grabación y capacidad.

SELECT

Crea un archivo seleccionando datos específicos de un archivo existente en disco flexible, el usuario especifica los registros que deben seleccionarse, los campos del archivo de salida, su contenido y el orden en que deben incluirse en los registros. La organización del archivo de entrada puede ser secuencial, relativa o indexada, mientras que la del archivo de salida es secuencial.

SORT GENERAL PARA DISCOS FLEXIBLES

Esta rutina clasifica datos procedentes de un archivo de entrada según la clave o claves (de 1 a 10) que se especifiquen, en orden ascendente o descendente, obteniéndose un archivo de salida clasificado.

El tamaño del archivo de entrada no debe ser superior al contenido de un disco flexible. La organización del archivo de entrada puede ser secuencial, relativa o indexada; los registros pueden ser de longitud fija o variable. La organización del archivo de salida puede ser secuencial o indexada según se especifique.

SORT ESPECIAL

Es una rutina especial para el mantenimiento de archivos indexados en disco flexible. Esta rutina inserta los registros añadidos al archivo durante una operación de salida, en el área de datos clasificados.

SUMLOG

Es una rutina para el servicio técnico que crea un archivo único a partir de los archivos de errores contenidos en varios discos flexibles.

HERRAMIENTAS DE SOFTWARE

Hay una serie de herramientas cuya función es auxiliar en el desarrollo del software de usuario y generación del sistema.

TEXT EDITOR

Permite al programador crear y/o editar archivos fuente de entrada al compilador COBOL o hileras de control. Un objetivo secundario es efectuar cambios y/o eliminaciones en un archivo de datos ya existente con registros de longitud fija o variable de hasta 80 caracteres de longitud.

COMPILADOR DE COBOL 74

Es un programa traductor que convierte un programa fuente escrito en COBOL DPS en un programa en lenguaje objeto. El programa objeto es almacenado en disco flexible y posteriormente copiado a un disco de aplicaciones. Después de ser cargado en memoria, el programa objeto es ejecutado mediante el intérprete COBOL.

USER DEBUG

Esta herramienta sirve de ayuda para la depuración de programas objeto COBOL. Comprende una serie de instrucciones que permiten parar la ejecución de un programa, presentar en pantalla y modificar el valor de campos, etc.

GENERADOR DEL SISTEMA (SYSGEN)

La rutina SYSGEN permite confeccionar un sistema operativo de acuerdo con la configuración hardware y software deseada. SYSGEN contiene un cuestionario que ayuda a la selección de las características deseadas en el sistema operativo.

AUTOPAT

Proporciona un medio de efectuar modificaciones de forma controlado, sobre rutinas de utilidad ya distribuidas.

LENGUAJE DE CONTROL DEL SISTEMA

En este documento se usa la forma abreviada de los verbos y parámetros de las instrucciones JCL (se adjunta una lista de abreviaturas).

NOTACIÓN

- Cuando se incluye entre corchete " [ ]", una porción del formato general, el programador tiene la opción de incluir u omitir esa porción.

  La superposición vertical de dos o más posibilidades dentro de los corchetes permite que el programador selecciones uno de esos formatos, que son excluyentes.

  Si no se especifica opción, se toma el primero por omisión.

- Cuando una porción de un formato general se incluye entre llaves "{ }", el programador tiene que seleccionar uno de los formatos apilados verticalmente dentro de las llaves.

- Una etiqueta seguida por un signo de dos puntos (:) y un comentario precedido por un punto y coma (;), pueden ser usados con cualquier instrucción.

ABORT

Abortar ..... Cancelar el proceso en curso
ABORT

ASSIGN

Asignar ..... Un archivo ya existente en disco flexible
ASSI Lfid Pfid (n,FD) [,AP] [,SCR]

Asignar ..... Un archivo nuevo de datos endisco flexible
ASSI Lfid Pfid (n,FD),NEW,ns [,AP] [,SCR] [,M] [,I]

Asignar ..... Un archivo nuevo que contendrá un programa objeto o una rutina de utilidad.
ASSI Lfid Pfid (n,FD),NEW,ns,AP

Asignar ..... Manipulador de tarjetas
ASSI Lfid (Ø,{LL / LU}),lpi,pl,pr

Asignar ..... Código de barras (los LL/LU correspondientes tienen que haber sido asignados previamente)
ASSI Lfid (Ø,{BL / BU}),pl,pr

Asignar ..... Manipulador de diario o el alimentador de formularios continuos.

ASSI Lfid (Ø,{C1 / C2}),,pl,pr

Asignar ..... Impresora de línea

ASSI Lfid (Ø,LP)

Asignar ..... Archivo en cassette de cinta magnética

ASSI Lfid Pfid (n,CA)

CHANGE

Cambiar ..... Identificación física del archivo

CHAN Pfid [(n,FD)] [TO] Pfid2

DELETE

Eliminar ..... Un solo archivo

DELE Pfid [(n,FD)]

Eliminar ..... Cualquier rutina de utilidad (excepto MOVE)

DELE Pfid [(n,FD)] ,SO

Eliminar ..... Todos los archivos no-software

DELE vs [(n,FD)]

Eliminar ..... Todos los archivos, incluyendo rutinas de utilidad (excluyendo O.S. y rutina de utilidad MOVE).

DELE vs [(n,FD)] ,SO

DISPLAY

Presentar ..... La etiqueta de un cassette

DISP Pfid [(n,CA)] [(CT) / (Ø,LP)]

Presentar ..... El directorio de un disco flexible

DISP [(n,FD)] [(CT) / (Ø,LP)]

Presentar ..... Los datos del directorio de un archivo

DISP Pfid [(n,FD)] [(CT) / (Ø,LP)]

Presentar ..... El archivo de errores

DISP SYSLOG [(n,FD)] [(CT) / (Ø,LP)]

Imprimir ..... Y borrar el archivo de errores

DISPLAY SIGLOG [(n,FD)] (Ø,LP),CLEAR

DUMP

Volcar ..... Un archivo

DUMP Pfid [(n,FD) / (n,CA)] [,b,e] [(CT) / (Ø,LP)] [,ASC / HEX / AEX / SBS]

Volcar .....Un área absoluta de un medio magnético

DUMP b,e [(n,FD) / (n,CA)] [(CT) / (Ø,LP)] [ASC / HEX / AEX / SBS]

EXECUTE

Ejecutar ..... Una rutina de inicializar discos flesibles

EXEC FLEXIT [(n,FD)]

Ejecutar ..... Copia de Overlays

EXEC OVCOPY (n,FD)

Ejecutar ..... Un programa objeto

EXEC Pfid [(n,FD)] [,DEBUG]

Ejecutar ..... Una hilera de control

EXEC Pfid [(n,FD)] [,NOLI]

Ejecutar ..... Rutina de utilidad de edición de textos

EXEC EDIT [(n,FD)] [,RE]

Ejecutar ..... Clasificación especial de archivos indexados

EXEC SPECSORT [(n,FD)]

Ejecutar ..... Rutina de utilidad de clasificación

EXEC SORT [(n,FD)]

IF

Si ..... El código JCL es igual a cualquier valor n

IF n_0 ..... n_7 [GO TO] Label-1

Si ..... El código JCL no es igual a n

IF NOT n [GO TO] Label-1

Si ..... El código JCL es cero

IF Ø [GO TO] Label-1

Si ..... El código JCL no es cero

IF NOT Ø [GO TO] Label-1

Si ..... El código JCL es menor que n

IF < n [GO TO] Label-1

Si ..... El código JCL no es menor que n

IF NOT < n [GO TO] Label-1

Si ..... El código JCL es mayor que n

IF > n [GO TO] Label-1

Si ..... El código JCL no es mayor que n

IF NOT > n [GO TO] Label-1

NOTAS: - La instrucción IF-SI, condicional, únicamente puede ser introducida en una hilera de control.

- Label-1, es la etiqueta de un comando <u>posterior</u> al que se encuentra en curso.

SET

Poner ..... Los indicadores de programa COBOL

SET SWIT n_1 ..... n_8 {ON / OFF}

Poner ..... El código JCL en n (n=Ø a 7)

SET JCL [TO] n

MOVE

Transferir ..... Un archivo

MOVE Lfid [TO] Lfid

Transferir ..... Una rutina de utilidad

MOVE Lfid [TO] Lfid,SO

Transferir ..... Un archivo indexado y omitir los registros eliminados

MOVE Lfid [TO] Lfid,PACK

Transferir ..... Todos los archivos no software

MOVE (n,[FD]) [TO] (n,[FD]),SO [,AU [(CT) / (Ø,LP)]]

Transferir ..... Todos los archivos incluyendo rutinas de utilidad (excluyendo O.S.)

MOVE (n, FD) TO (n,FD),SO [,AU [(CT) / (Ø,LP)]]

Transferir ..... Un archivo por registros a un dispositivo de presentación visual

MOVE Lfid [,b,e] {Lfid / CT} [ASC / HEX / AEX / SBS]

NOTA: Tienen que haber sido asignados previamente los nombres lógicos de los archivos.

QCOPY

Copiar ..... Un disco flexible

QCOPY (n,FD) [TO] (n,FD)

ABREVIATURAS PARA EL JCL

AEX = ASCII y Hexadecimal

AP = Aproximado

ASC = ASCII

AU = Con listado

b = Comienzo sector/bloque - número de registros relativo en MOVE
primero = 1
- número de bloque relativo en DUMP
primero = Ø

BL = Código de barras inferior (derecha)

BU = Código de barras superior (izquierda)

C1 = Diario o alimentador de impresos continuos 1

C2 = Alimentador de impresos continuos 2

CA = Cassette

CT = Pantalla (CRT)

e = Fin de sector/bloque - número de registro relativo desde 1 en MOVE
- número de bloque relativo desde Ø en DUMP

FD = Disco flexible

HEX = Hexadecimal

I = Tipo de intercambio estándar (EBCDIC)

JCL = Lenguaje de Control de Trabajos

Label = Nombre de referencia a la que se ha de acceder mediante una instrucción IF. La referencia puede ser entre 1 y 6 caracteres, inmediatamente seguidos por el signo de dos puntos (:).

Lfid = Identificación lógica del archivo (1-10 caracteres)

LL = Manipulador de tarjetas inferior (derecha)

LP = Impresora de líneas

Lpi = Líneas por pulgada (2,54 cms.) (4, 5, 6 u 8 - supuesto por omisión 6)

LU = Manipulador de tarjetas superior (izquierda)

M = Archivo multivolumen en FD

n = Número de la unidad (supuesto por omisión Ø)

NEW = Nuevo

NO = Nuevo objeto para la versión 2.0
Sin control de lectura después de escritura, empezando con la versión 3.0

NOLI = Sin listado

ns = Número de sectores (número par)

Pfid = Identificación física del archivo (1-8 caracteres para FD; 1-10 caracteres para CA)-optativo si es idéntico a Lfid.

pl = posición izquierda

pr = posición derecha ver tabla

RE = Recuperar

SBS = HEX y ASCII lado a lado - no para CRT.

SCR = Transitorio (para eliminar al ser cerrado)

SO = Software

SWIT = Indicador

vs = Número de serie

NORMAS PARA ASIGNACION DE 4501

Tipo de Dispositivo	12 caracteres/pulgada posición izq.	12 caracteres/pulgada posición dcha.	10 caracteres/pulgada posición izq.	10 caracteres/pulgada posición dcha.
C1	7-228	máx. 270	6-190	máx. 225
C2	7-228	máx. 270	6-190	máx. 225
LU	7-208	máx. 270	6-173	máx. 225
LL	78-208	máx. 270	65-173	máx. 225
BU	21-250	máx. 264	18-205	máx. 220
BL	80-250	máx. 264	67-205	máx. 220

NORMAS PARA ASIGNACION DE 4504

C1 : $1 \leq pi \leq 132$
$pi < pd \leq 174$
$lpi = 6$

LL : $1 \leq pi \leq 125$
$pd = 174$
$lpi = 4, 5, 6 \text{ u } 8$

BL : $22 \leq pi \leq 151$
$pd \leq 167$

En 4504 no se dispone de C2, LU, BU

COMPILADOR COBOL

GENERALIDADES

La misión de un compilador COBOL es generar un programa objeto a partir de un fuente, para ser traducido posteriormente a lenguaje máquina por el COBOL RUNTIME INTERPRETER.

El compilador consiste en tres módulos con los nombres físicos de COBOL, COMPIL y CAREPK, que deben estar contenidos en un mismo diskette.

El programa fuente puede estar contenido en cassette o disco mientras que el programa objeto se obtendrá en disco.

La entrada al compilador, como ya se ha dicho, es el programa fuente, mientras que las salidas serán el programa objeto por una parte y por otra el listado de compilación, que contiene las líneas fuente, errores y mapa de memoria para proporcionar una referencia al ejecutar los programas con DEBUG.

ROLL DE MEMORIA

Para la compilación se necesita un cierto área de memoria, que dependerá del número de líneas fuente del programa, así como de las referencias y literales definidos por el usuario.

Mediante la siguiente fórmula se puede obtener una aproximación acerca de las necesidades de memoria para un programa determinado, con unas tolerancias de aproximadamente un 10%.

$$S = \frac{N + U\ (10+2W/3) + L\ (6+C/2)}{500}$$

Los distintos parámetros que aparecen en la fórmula son:

N = Número de líneas fuente

U = Número de referencias definidas por el usuario

W = Número medio de caracteres por referencia

L = Número de literales

C = Número medio de caracteres por literal

S = Tamaño necesario de memoria en K bytes.

Las siguientes cifras se refieren al supuesto de un programa de 700 líneas, y se pone de manifiesto el gran ahorro de memoria que puede obtenerse actuando sobre el número de referencias y la longitud de las mismas. Estos, y en este orden, habrán de ser los parámetros sobre los que debe actuar el programador, pues las restantes, salvo pequeñas diferencias, vendrán impuestas por la necesidad del programa.

N =	700	700	700	
U =	100	100	100	
W =	9	4	4	
L =	100	100	100	
C =	14	14	14	
S =	7,2	6,5	5,3	($\pm$ 10%)

INSTRUCCIONES DE OPERACIÓN

Las instrucciones de operación que debe utilizar el operador para la compilación de un programa son las siguientes:

1. Asignación del archivo que contiene el programa fuente con el nombre lógico SI

   ASSIGN SI TO id-fis (n,dt)

   donde dt puede ser o bien CA (cassette) o DI (disco) o FD (diskette)

2. Asignación del archivo que contendrá el programa objeto con el nombre lógico BO.

   ASSIGN BO TO id-fis (n,dt), NEW/OLD ,ns,AP

   donde dt únicamente puede ser DI (disco) o FD (disco flexible)

3. Asignación del listado de compilación con el nombre lógico LO

   ASSIGN LO TO (n,LP)

   En caso de no efectuarse esta asignación la compilación será por pantalla.

4. Cargar el compilador COBOL en memoria mediante

   EXECUTE COBOL (n,FD)

   con lo que al pulsar FT empieza a efectuarse la compilación.

DEPURACION DE PROGRAMAS (USER DEBUG)

GENERALIDADES

Este módulo de software proporciona funciones de diagnóstico para encontrar problemas de programación durante la ejecución de programas de aplicación.

Se sitúa en memoria al tiempo que el programa objeto, mediante el parámetro DEBUG incorporado a la instrucción EXECUTE.

```
EXECUTE pfid (unit, dt), DEBUG
```

donde pfid es el nombre físico del archivo que contiene el programa a ejecutar.

Completada la carga en memoria, y cada vez que se acceda al USER DEBUG, se presenta en pantalla:

```
DEBUG = XXYYYY pfid
D?
```

donde XXYYYY indica la posición de la siguiente instrucción a ser ejecutada; XX es el número de overlay, YYYY es la situación relativa de la instrucción objeto dentro del overlay.

Esta situación para cada instrucción puede encontrarse en el listado de compilación en la columna DEBUG.

D? indica una petición por parte del sistema de una instrucción del USER DEBUG, que habrá de ser introducida por teclado; cada una de estas instrucciones se terminará mediante FA, con lo que el sistema pasa a ejecutarla; si hubiera un error en la instrucción aparecería ?, con lo que se puede introducir una nueva instrucción que anula a la anterior.

Introduciendo A, seguido de FA, acaba la utilización del USER DEBUG; si se quisiera utilizar de nuevo habría que cargar otra vez el programa objeto.

A continuación se exponen las distintas instrucciones del USER DEBUG.

INSTRUCCIÓN DE DIRECCIÓN PARADA (ADDRESS STOP)

El formato de esta instrucción es AXXYYYY

Su función es ejecutar el programa hasta la instrucción cuya posición viene especificada por XXYYYY, parándose en esta posición y dando opción para introducir una nueva instrucción DEBUG.

Si se introduce una A, sin más, se termina de efectuar el DEBUG y sigue ejecutándose el programa normalmente.

Si no se introduce XX se supone que el número de overlays es el 00.

Al especificar YYYY, pueden eliminarse los ceros a la izquierda.

INSTRUCCIÓN DE PASO A PASO (SINGLE STEP)

Es formato de esta instrucción es S

su función es ejecutar un paso de programa y para la ejecución, introduciendo FA sucesivamente se van ejecutando los pasos de programa uno a uno hasta que se introduzca una instrucción de de tipo A.

INSTRUCCIÓN DE PRESENTACIÓN DE SITUACIÓN (DISPLAY LOCATION)

El formato de esta instrucción es L

su función es presentar la situación relativa de la siguiente instrucción objeto COBOL a ser ejecutada; se utiliza cuando se borra la pantalla y por tanto, también la situación.

INSTRUCCIÓN PARA INSERCIÓN DE TRAMPAS (TRAP INSERT)

El formato de esta instrucción es T XXYYYY

su función es instalar una trampa en la dirección especificada por XXYYYY. Cuando se accede a una trampa la ejecución del programa se para automáticamente y la trampa se elimina. Puede haber en el programa hasta 10 trampas activas simultáneamente.

INSTRUCCIÓN PARA ELIMINAR TRAMPAS (REMOVE TRAP)

El formato de esta instrucción es R

su función es eliminar la última trampa que se haya insertado.

INSERCIÓN PARA PRESENTAR DATOS (DISPLAY DATA)

El formato de esta instrucción es: D aaaa,bbbb,tttt

su función es presentar un campo dato residente en memoria; el campo viene definido por los tres parámetros siguientes, obtenidos de las tres primeras columnas del mapa de memoria de la Data división en el listado de compilación.

. aaaa, especifica la localización hexadecimal del campo a presentar, respecto del principio del programa, se encuentra en la columna ADDRESS.

. bbbb, especifica el tamaño en bytes del campo hexadecimal, se encuentra en la columna SIZE.

. tttt, especifica el tipo de dato, se encuentra en la columna DEBUG. Este parámetro puede ser uno de los siguientes:

ABS = Hilera alfabética
ANS = Hilera alfanumérica
ANSE = Hilera alfanumérica editada
BIT = Booleano
GRP = Grupo
HEX = Hexadecimal
NBU = Numérico binario sin signo
NPS = Numérico compactado con signo
NPU = Numérico compactado sin signo
NSE = Hilera numérica editada
NSS = Hilera numérica con signo
NSU = Hilera numérica sin signo

El dato se presenta en la forma ttttaaaaddddd.... donde dddd... es el dato seleccionado; los datos mayores de 24 bytes se presentan en varias líneas.

INSTRUCCIÓN PARA PRESENTAR PARÁMETROS

El formato de esta instrucción es Pnnnn,aaaa,bbbb,tttt

su función es presentar un parámetro, entendiéndose por tal un campo de la LINKAGE SECTION de la DATA DIVISION de un programa ejecutado por la instrucción COBOL CALL. Para poder ser presentado, el dato debe encontrarse en memoria y por tanto, el programa llamado se debe estar ejecutando.

La entrada nnnn especifica el número de parámetro y las aaaa, bbbb y tttt tienen la misma definición que en el caso anterior, se encuentran en las columnas ADDRESS, SIZE y DEBUG, respectivamente, del listado de compilación.

INSTRUCCIÓN PARA CAMBIO DE DATOS (CHANGE DATA)

El formato de esta instrucción es Caaaa,bbbb,tttt,dddd.....

Las entradas aaaa, bbbb y tttt tienen la misma definición que las correspondientes de la instrucción D. La entrada dddd... es el valor del dato a insertar en la localización especificada, el número de caracteres del mismo ha de ser tal que la línea de entrada no tenga más de 80.

Si se introducen menos bytes que lo especificado en la longitud del campo, se alinean a la izquierda con relleno de espacios para los tipos ABS, ANS, ANSE, GRP, NSE, NSS y NSU.

Si el número de bytes es mayor, el dato queda truncado por la derecha.

Para los tipos BIT, HEX, NBU, NPS o NPU el dato se alinea a la derecha y se rellena con ceros; si el número de bytes es mayor queda truncado por la izquierda.

Si el tipo de dato es BIT o HEX, pueden insertarse espacios en la entrada de datos para facilitar la lectura, al llevar los datos a memoria se ignoran los espacios.

Si el tipo de dato introducido en la instrucción es diferente que el tipo de dato especificado en el listado de compilación puede fallar el sistema.

INSTRUCCIÓN PARA CAMBIO DE PARÁMETROS

Esta instrucción tiene el formato CP nnnn,aaaa,bbbb,tttt,dddd...

donde nnnn, aaaa, bbbb y tttt tienen la misma definición que en la instrucción P, y dddd... es el dato a insertar.

Deben tenerse en cuenta las mismas especificaciones que para la instrucción C.

INSTRUCCIÓN DE PARADA EN PROGRAMA LLAMADO (CALLED PROGRAM STOP)

El formato de esta instrucción es @ pfid

su función es detener la ejecución en la primera instrucción del programa especificado por pfid.

Cualquier detención del programa desactiva la instrucción @ . Si una instrucción de los tipos A, S ó T pasan la ejecución del programa antes de acceder a la localización especificada en @ , ésta deja de ser efectiva.

RUTINA DE EDICION DE TEXTOS (TEXT EDITOR)

GENERALIDADES SOBRE EDIT

La rutina de utilidad EDIT (TEXT EDITOR), proporciona el medio de crear y cambiar:

1. Archivos de entrada al compilador COBOL (Programas Fuente)

2. Hileras de control y archivos de parámetros

3. Archivos de datos

Para utilizar EDIT hay que tener en cuenta los siguientes puntos:

- Los archivos de entrada y salida pueden tener registros de longitud fija o variable, sin embargo la longitud máxima de los registros es de 80 bytes.

- Los archivos de entrada y salida utilizados por EDIT pueden estar contenidos en disco o cassette. El teclado también puede utilizarse como medio de entrada.

- Los archivos de entrada a EDIT han de tener organización secuencial; los archivos de salida de EDIT tienen organización secuencial.

ARCHIVO DE TRABAJO

Todas las instrucciones de EDIT actúan sobre el contenido de un archivo de trabajo, que deberá ser creado con anterioridad a cargar EDIT en memoria, mediante el comando JCL ASSIGN.

Este archivo de trabajo se llenará posteriormente con el contenido del archivo a modificar, mediante las instrucciones CONCAT o INSERT, quedanto este intacto.

Cuando EDIT se termina normalmente, mediante la instrucción QUIT o cuando sucede un error de entrada/salida el contenido del archivo de trabajo puede recuperarse mediante la instrucción:

```
EXECUTE EDIT  (n,dt), RE
```

Si EDIT termina de forma anormal, el archivo de trabajo se considera vacio al intentar su recuperación.

LINEA EN CURSO

La línea en curso es el registro sobre el que se ejecutan todas las funciones de EDIT. La nueva línea en curso aparece a continuación de ser ejecutada cualquier función de EDIT.

El registro que sigue a la línea en curso es el siguiente del archivo a menos que no siga ninguno a la línea en curso.

Cuando EDIT accede al archivo de trabajo utiliza la línea en curso para determinar la posición del registro a ser accedido; dependiendo de las instrucciones y parámetros incluidos en las mismas, EDIT accede a una de las siguientes localizaciones en el archivo de trabajo:

- La situación especificada por la línea en curso (como en una instrucción .CHANGE, cuando no se especifica ninguna posición en la misma).
- La situación que sigue a la especificada en la línea en curso (como en una instrucción CONCAT).
- Una posición relativa a la de la línea en curso (como en una instrucción POSITION).
- La posición especificada en una instrucción EDIT (como en la instrucción CHANGE, cuando la situación se especifica en la misma).

MODOS DE ACCESO A LOS REGISTROS

Hay dos formas posibles de acceso a los registros del archivo de trabajo:

a) MODO DE ACCESO RELATIVO

El modo de acceso relativo accede a los registros de cualquier archivo de entrada o del archivo de trabajo. En la instrucción de EDIT se debe introducir un valor de 1 a 65535, que identifica cualquier registro por su posición relativa al primero del archivo. Debe incluirse un asterisco en la instrucción cuando se utiliza este método de acceso.

b) MODO DE ACCESO SECUENCIAL

Cuando los registros de un archivo de entrada, o del archivo de trabajo, contienen números de secuencia, y los registros están clasificados en secuencia ascendente, puede utilizarse el método de acceso secuencial. Un número de secuencia de 6 dígitos en una instrucción de EDIT identifica un registro del archivo. Para que los resultados en la búsqueda de un determinado registro sean los esperados, los registros del archivo deben estar en la secuencia adecuada.

Dependiendo del registro correspondiente a la línea en curso, la búsqueda se realiza en secuencia ascendente o descendente. Esta búsqueda acaba cuando tiene lugar una de las siguientes condiciones:

. Se localiza el registro con el número de secuencia especificado.

. Se localiza un registro con número de secuencia superior al del buscado, cuando la búsqueda se realiza en orden ascendente.

. Se localiza un registro con número de secuencia inferior al del buscado, cuando la búsqueda se hace en orden descendente.

. Se accede al final o al principio del archivo.

PROCEDIMIENTOS DE ASIGNACIÓN

Antes de cargar EDIT en memoria hay que efectuar las asignaciones adecuadas. Estas son:

a) Archivo o archivos de entrada: Si el archivo de entrada se introduce por teclado, no es necesaria ninguna asignación. Sin embargo, cuando se requieren 1 o más archivos de entrada, hay que efectuar las asignaciones correspondientes a todos ellos mediante las instrucciones de la forma:

ASSIGN id-log TO id-fis (n,dt)

b) Archivo de salida: Sobre el archivo de salida se vuelca el contenido del archivo de trabajo después de acabadas las funciones de EDIT. Si el contenido del archivo de trabajo ha de ser volcado en un archivo de entrada, no es necesaria ninguna instrucción ASSIGN para archivo de salida. Si el archivo de salida no coincide con uno de entrada, es necesaria una instrucción de la forma:

ASSIGN id-log TO id-fis (n,dt), $\begin{matrix}\text{NEW}\\\text{OLD}\end{matrix}$, ns

c) Archivo de trabajo: El archivo de trabajo, independientemente de si es viejo o nuevo, debe tener el nombre lógico EWF, y debe asignarse mediante la instrucción:

ASSIGN EWF TO id-fis (n,dt), $\begin{matrix}\text{NEW}\\\text{OLD}\end{matrix}$, ns

d) Impresora de línea: Si es necesaria una copia impresa del contenido del archivo de trabajo debe asignarse una impresora con el nombre lógico LO. La impresora, si fuere necesario, se asignará con una instrucción de la forma:

ASSIGN LO TO (n,LP)

INSTRUCCIONES DE EDIT

Las siguientes instrucciones constituyen los distintos medios de incorporar, cambiar o eliminar registros. Solamente son necesarios los dos primeros caracteres de cada instrucción, así si se necesita la instrucción DELETE sólo es necesario especificar DE.

CHANGE

Esta instrucción se utiliza para modificar el contenido de un registro, los cambios pueden realizarse una vez que el registro está presentado en pantalla.

Cualquier carácter del registro puede ser duplicado mediante la tecla ( ), cualquier otro carácter que se introduzca sustituye al correspondiente del contenido del registro.

Si se pulsan las teclas ( ) o (FA), antes de introducir ningún carácter, el contenido del registro no se modifica y no se introduce ningún dato en el archivo de trabajo.

A continuación hay tres formatos distintos de la instrucción CHANGE y las funciones que realizan.

CHANGE

Esta instrucción presenta en pantalla la línea en curso y hace que los siguientes caracteres, hasta 80, que se introduzcan, sustituyan el contenido del registro. La línea en curso sigue siendo la misma.

CHANGE n

Esta instrucción hace que se presente en pantalla el registro con número de secuencia n, sobre el que se podrán hacer las modificaciones adecuadas. La línea en curso pasa a ser la especificada por el número de secuencia n.

CHANGE *n

Esta instrucción presenta en pantalla el registro cuya posición relativa viene especificada por n, sobre el que se podrán hacer los cambios pertinentes. La línea en curso pasa a ser el registro con número relativo n.

CONCAT

Esta instrucción inserta en el archivo detrabajo, a partir de la línea en curso, los registros seleccionados contenidos en el archivo especificado por id-log. El archivo debe tener oganización secuencial, registros de longitud fija o variable con una longitud máxima de bloque de 512.

Cuando el archivo de entrada se encuentra en cassette, se necesita una entrada adicional, que especifique las características del archivo en cinta magnética. El formato de esta entrada es como sigue:

CO id-log (p,q)

- La "p" especifica la longitud del registro cuando el archivo contiene registros de longitud fija. Si no se especifica p, se suponen registros de longitud variable, con una longitud máxima de 80 caracteres.
- La "q" especifica el tamaño del bloque; se admite una longitud máxima de 512 caracteres. Si no se especifica q, el archivo se se supone con registros de longitud fija, del tamaño especificado por p, y tamaño de bloque el mayor múltiplo de p, menor o igual que 512.

Se describen 11 instrucciones CONCAT distintas, así como las funciones que desempeñan cada una de ellas. La línea en curso, después de ejecutar cada instrucción, pasar a ser el último registro introducido en el archivo de trabajo.

CONCAT id-log

Esta instrucción inserta en el archivo de trabajo, a continuación de la línea en curso, todos los registros contenidos en el archivo asignado con el identificador lógico que se especifica.

CONCAT id-log, m

Esta instrucción inserta en el archivo de trabajo, a continuación de la línea en curso, los registros contenidos en el archivo asignado con el identificador lógico que se especifica, desde aquel que tiene número de secuencia m, hasta el final del archivo.

CONCAT id-log, *m

Inserta en el archivo de trabajo, a continuación de la lína en curso, los registros contenidos en el archivo asignado con el identificado lógico que se especifica, desde aquel que ocupa la posición relativa m, incluido hasta el final del archivo.

CONCAT id-log, m, n

Inserta en el archivo de trabajo los registros comprendidos entre los números de secuencia m y n, estos incluidos.

CONCAT id-log, *, m

Inserta en el archivo de trabajo todos los registros anteriores al de posición relativa m, incluido éste.

CONCAT id-log, *m,n

Inserta en el archivo de trabajo todos los registros comprendidos entre los números relativos m y n, incluyendo éstos.

CONCAT id-log (p,q)

Esta instrucción inserta en el archivo de trabajo, a continuación de la línea en curso, todos los registros contenidos en el

archivo con el identificador lógico que se especifica. Si el archivo contiene registros de longitud fija, p especifica la longitud del registros. Si no se hace ninguna entrada para p, el archivo se supone con registro de longitud variable con una longitud máxima de 80 bytes. La q especifica el tamaño del bloque y no debe ser mayor de 512 caracteres. Si no se hace ninguna entrada para q, el archivo tiene registros de longitud fija, especificada por p, y el tamño del bloque es el mayor múltiplo de p menor o igual que 512.

```
CONCAT id-log (p,q),m

CONCAT id-log (p,q),*m

CONCAT id-log (p,q),*,m

CONCAT id-log(p,q),m,n

CONCAT id-log (p,q),*m,n
```

Estas instrucciones tienen el mismo significado que sus correspondientes sin especificar (p,q).

DELETE

Esta instrucción se utiliza para eliminar un registro o varios del archivo de trabajo. Una vez eliminados los registros especificados las funciones ejecutadas por EDIT dependen de:

1. Si hay registros en el archivo de trabajo que sigan al último registro eliminado.

2. Si todos los registros del archivo de trabajo han sido eliminados, las posibles condiciones que pueden resultar son:

   . Si hay registros que sigan a los eliminados, EDIT desplaza el resto del archivo hasta eliminar huecos, de forma que el primer registro disponible sustituya al primero eliminado. La situación relativa del registro eliminado se mantiene como línea en curso.

   . Si no hay ningún registro que siga a los eliminados, EDIT determina si todos los registros del archivo de trabajo se han eliminado.

     - Si el archivo de trabajo está vacio, la línea en curso se pone a ceros.

     - Si hay registros en el archivo de trabajo la situación relativa del último registro del archivo se toma como línea en curso.

Una vez acabado DELETE se presenta en pantalla el contenido del registro especificado por la línea en curso, siempre y cuando el archivo de trabajo no se haya quedado vacio.

A continuación se describen cinco instrucciones distintas de DELETE.

DELETE

Esta instrucción elimina la línea en curso.

DELETE n

Esta instrucción elimina el registro con número de secuencia n.

DELETE *n

Esta instrucción elimina el registro con el número relativo n.

DELETE m,n

Elimina todos los registros con números de secuencia comprendidos entre m y n, incluyendo éstos.

DELETE *m,n

Elimina todos los registros con números relativos comprendidos entre m y n, incluyendo éstos.

DISPLAY

Esta instrucción presenta en pantalla los registros seleccionados. El último registro presentado pasa a ser la nueva línea en curso.

Hay cinco formatos distintos para esta instrucción:

DISPLAY

Presenta en pantalla la línea en curso.

DISPLAY n

Presenta el registro con número de secuencia n.

DISPLAY *n

Presenta el registro con número relativo n.

DISPLAY m,n

Presenta todos los registros comprendidos entre los números de secuencia m y n, incluyendo éstos.

DISPLAY *m,n

Presenta todos los registros comprendidos entre los de número

relativo m y n, incluyendo éstos.

FIND

Esta instrucción busca un registro determinado y lo presenta como registro de línea en curso. Hay dos formatos diferentes para la instrucción FIND.

FIND n

Esta instrucción localiza y presenta el registro con número de secuencia n.

FIND *n

Localiza y presenta el registro con número relativo n.

INSERT

Esta instrucción proporciona un método para hacer entradas por teclado al archivo de trabajo, a continuación de la línea en curso. Cada registro introducido se acaba pulsando FA.

Para acabar la instrucción INSERT, introdúzcase (\) o (FA) como primer carácter del registro de entrada.

El siguiente registro accesible será la nueva línea en curso. A continuación se exponen dos formatos distintos de INSERT.

INSERT *D

Esta instrucción permite entrada por teclado de un registro a continuación de la línea en curso. La D es opcional, y si se especifica, se proporciona una duplicación de caracteres, como en la instrucción CHANGE.

INSERT D

Esta instrucción permite entrada por teclado de un registro a continuación de la línea en curso, todos los registros introducidos se desplazan seis posiciones a la derecha, con truncamiento a la derecha, si fuera necesario para mantener la longitud de 80 caracteres. El campo de seis caracteres a la izquierda se rellena con espacios.

MOVE

Esta instrucción copia una o más líneas fuente a una posición en el archivo de trabajo. Los registros entre m y n, incluyendo estos dos, se mueven a la posición siguiente a la especificada por x. Si se introduce DELETE, los registros entre m y n, incluyendo estos dos son eliminados.

El último registro movido, en su nueva posición, pasa a ser la nueva línea en curso.

MOVE DELETE *m,n-x

Esta instrucción transfiere los registros entre la posición relativa m y n, incluyendo ambos, a la posición siguiente a la especificada por x. Si se incluye DELETE se suprimen los registros originarios.

MOVE DELETE m,n-x,y

Esta instrucción transfiere los registros con números de secuencia comprendidos entre m y n, incluyendo ambos, a la posición que sigue inmediatamente a x. Los números de secuencia de los registros transferidos se aumentan en el valor especificado en y, empezando con el de x; si no se hace ninguna entrada en y, los números de secuencia se aumenta de 10 en 10. Si no se especifica n, se transfiere un solo registro. Si se especifica DELETE, se eliminan los registros de origen.

POSITION

Esta instrucción localiza un registro determinado y lo presenta como línea en curso.

Hay dos formatos para la instrucción POSITION:

POSITION n

Esta instrucción localiza y presenta el registro que se encuentra n posiciones por encima de la línea en curso.

POSITION -n

Esta instrucción localiza y presenta el registro que se encuentra n posiciones por debajo de la línea en curso.

PRINT

Esta instrucción imprime una o más líneas fuentes del archivo de trabajo; en las líneas fuentes se especifican por n o por m y n. Antes de efectuar la instrucción PRINT, debe asignarse el archivo de impresión con nombre lógico LO.

El último registro impreso pasa a ser la nueva línea en curso.

A continuación se indican cuatro formatos distintos para la instrucción PRINT:

PRINT n

Esta instrucción imprime el registro con número de secuencia n.

PRINT *n

Esta instrucción imprime el registro con número relativo n.

PRINT m,n

Imprime todos los registros con número de secuencia comprendidos entre m y n, incluyendo estos dos.

PRINT *m,n

Imprime todos los registros con números relativos comprendidos entre m y n, incluidos ambos.

QUIT

Esta instrucción da por finalizado EDIT y devuelve el control al sistema operativo.

RESEQUENCE

Esta instrucción cambia los primeros seis caracteres de cada registro en una serie de registros o en todos los del archivo de trabajo.

A continuación se exponen los dos formatos de RESEQUENCE:

RESEQUENCE x,y

Sitúa el número x (seis caracteres) en las posiciones 1 a 6 del registro número 1, y va incrementando la secuencia de todos los demás en el valor especificado por y. Si no se hace ninguna entrada para y, el valor de x se va aumentando de 10 en 10 y no puede ser mayor de 65535.

RESEQUENCE m,n-x,y

Cambia los números de secuencia de los registros comprendidos entre m y n, incluyendo ambos. La x indica el primer número de secuencia nuevo; la y el incremento que debe darse a los números de secuencia. Si no se especifica ningún valor para y, los incrementos se hacen de 10 en 10.

SAVE

Esta instrucción hace que los registros contenidos en el archivo de trabajo se graben en el archivo con el identificador lógico especificado.

El archivo de salida tiene organización secuencial y puede estar contenido en disco o cassette.

A continuación se describen los dos formatos para la instrucción SAVE.

SAVE id-log

Esta instrucción hace que los registros contenidos en el archivo de trabajo se graben en el archivo especificado. El archivo de salida se crea con registro de longitud variable con un tamaño máximo de registros 80. El archivo tiene un tamaño de bloque de 512.

SAVE id-log (p,q)

Esta instrucción hace que los registros contenidos en el archivo de trabajo se graben en el archivo especificado. Los registros se crean de longitud fija, con la longitud especificada por p y tamaño de bloque especificado por q. Si no se especifica ningún valor para q, el archivo se considera de longitud fija y del tamaño especificado por p y el tamaño de bloque es el mayor múltiplo de p, menor o igual que 512. Si no se especifica ningún valor para p, el archivo se considera de registros de longitud variable con una longitud máxima de 80 y tamaño de bloque especificado por q.

SEARCH

Esta instrucción compara una hilera determinada con cada registro del archivo de trabajo, hasta que se encuentre un registro que contenga esa hilera. Una vez se encuentre este registro se presenta en pantalla y pasa a ser la nueva línea en curso.

SEARCH 'hilera'

Esta instrucción busca la hilera en secuencia ascendente a partir de la línea en curso, o incluyendo el registro de ésta.

SEARCH 'hilera',n

Hace la misma función que la anterior instrucción, pero limitando la búsqueda al número de registros especificados por n.

STRING

Esta instrucción permite cambiar la hilera-1, si se encuentra en el registro inspeccionado,por la hilera-2

STRING 'hilera-1','hilera-2'

Sustituye el valor hilera-1 en la línea en curso por el valor hilera-2.

STRING 'hilera-1','hilera-2',ALL

Sustituye el valor hilera-1 por el hilera-2, todas las veces que se encuentre en el archivo de trabajo.

STRING 'hilera-1','hilera-2',n

Sustituye el valor hilera-1 por el hilera-2, todas las veces que se presente en un número de registros especificado por n, a partir de la línea en curso.

INICIALIZACION DE DISCOS FLEXIBLES (FLEXIT)

GENERALIDADES SOBRE FLEXIT

La rutina de utilidad FLEXIT tiene dos funciones distintas:

a) Inicializar un disco flexible, con lo que se asigna una dirección de sector y pista únicas a cada uno de los sectores del disco, se verifican todas las áreas de datos por si hay zonas defectuosas, se asigna una pista alternada si se encuentra un un sector defectuoso, se graba una cabecera de volumen (sector 07), mapa de errores (05) y directorio del disco flexible.

   El directorio para un disco de densidad sencilla, se extiende desde el sector 8 al 19, y para uno de doble densidad desde el 8 al 52.

b) Cambiar la cabecera del volumen, que permite modificar el número de serie y la identificación del usuario. Esta función no destruye ningún archivo del disco flexible.

EJECUCIÓN DE FLEXIT

Una vez ejecutada la rutina de principio de día, se carga FLEXIT en memoria mediante la instrucción:

```
EXECUTE FLEXIT (n,FD)
```

en la unidad n debe estar instalado un disco flexible que contenga la rutina de utilidad FLEXIT.

Aparece en pantalla el siguiente menú:

```
ITEM   PARAMETER-IDEM                VALUE

 (1)   VOLUME SERIAL NUMBER          000000
 (2)   USER ID
 (3)   UNIT (1-3)                        1
 (4)   REBLD VOLHDR ONLY (Y/N)           N
 (5)   DENSITY (S/D/Q)                   D
 (6)   CERTIFY NBR. (1-9)                1)
 (7)   APLICATION FLEX (Y/N)             N

          WISH TO CHANGE AN ITEM (1-7/N) *
```

Los valores preasignados para cada uno de los parámetros, cuyo significado se explica a continuación, pueden modificarse anotando el número de parámetro y a continuación el valor deseado seguido de FA.

Una vez conseguido los valores adecuados se respondera N a la pregunta "WISH TO CHANGE AN ITEM (1-7/N) *", con lo que se ejecuta la rutina.

Los significados de los distintos parámetros y los valores aceptables son los siguientes:

a) VOLUME SERIAL NUMBER

. Debe introducirse el número de serie (Ø a 6 dígitos), que identificará el disco flexible. A continuación pulsar FA.

. Si no se introduce ningún valor antes de pulsar FA queda asignado como número de serie ØØØØØØ.

b) USER ID

. Hay que introducir de Ø a 14 caracteres de identificación del propietario del disco flexible. A continuación pulsar FA.

. Si no se hace ninguna entrada antes de pulsar FA, el campo se rellena con espacios.

c) UNIT

. Hay que introducir el número de unidad en que está instalado el disco flexible a inicializar, entradas válidas son Ø y 1. A continuación pulsar FA.

. Si no se introduce ningún valor antes de pulsar FA, se supone la unidad Ø.

d) CHANGE VOLUME HEADER Y/N

. Si únicamente se desea cambiar la cabecera, incorporando los parámetros antes introducidos, pulsar Y y FA. Esto no destruirá ningún archivo que pudiera estar contenido en el disco.

e) DENSITY

. Hay que especificar si el disco ha de ser inicializado como de densidad doble, sencilla o cuádruple, por lo cual hay que introducir una D, una S o una Q respectivamente. A continuación pulsar FA.

. Si no se introduce nada nates de pulsar FA el disco se inicializará como de doble densidad.

f) NO. OF CERTIFICATIONS

. Especifica el número de veces que hay que comprobar los sectores de datos antes de darlos por buenos. Hay que introducir un dígito de 1 a 9. A continuación pulsar FA.

. Si no se introduce nada antes de pulsar FA, el disco se inicializará como de datos (N).

Una vez terminada la inicialización aparece en pantalla:

END FLEXIT

Si hay que inicializar más discos flexibles, introducir R y pulsar FA. Continúese en el paso 2.

Si no hay más discos a inicializar introducir X y pulsar FA, con lo que se acaba FLEXIT.

OPERATORIA Y FUNCIONAMIENTO DE LA RUTINA DE UTILIDAD OVCOPY

La rutina de utilidad OVCOPY para el Release 5 se ha modificado respecto de la misma rutina utiliza en releases anteriores.

En los puntos siguientes se describen los pasos a seguir para ejecutar OVCOPY, así como las funciones que se ejecutan en cada uno de ellos.

- Una vez efectuada la rutina de principio de día y con el disco flexible, MASTER, instalado en la unidad número Ø, se debe cargar OVCOPY mediante la siguiente instrucción:

  EXECUTE OVCOPY

- Al aparecer el mensaje "1,FD MOUNT MEDIA", debe instalarse el disco flexible de datos o de aplicaciones que hay que actualizar en la unidad 1 y a continuación pulsar FA.

  NOTA: A fin de prevenir una posible destrucción de datos debe hacerse una copia de seguridad antes de llevar a cabo la actualización.

- Si el disco que se instala en la unidad 1 no es de datos ni de aplicaciones, aparecerá el mensaje "1,FD BAD MEDIA". Debe substituirse el disco de la unidad 1 por uno de datos o aplicaciones y pulsarse FA.

- En este punto la rutina empieza a copiar los overlays del sistema y si el disco es de aplicaciones, también el sistema operativo residente. En caso de que el nuevo sistema operativo ocupase mayor número de sectores que el antiguo aparecerá el mensaje:

  1, FD DPSSFOPS INSUF ROOM

  En este caso debe finalizarse la rutina pulsando X y a continuación FA. Para llevar a cabo la actualización deberá inicializarse un disco, efectuar OVCOPY y transferir todos los archivos de usuario y rutinas de utilidad a este disco mediante MOVE. Después de esto debe comenzarse de nuevo en el paso primero, para actualizar las rutinas de utilidad.

- A continuación, el sistema almacena en una tabla los nombres de todas las rutinas de utilidad contenidas en el disco que está instalado en la unidad 1.

- Se borran las rutinas de utilidad y los archivos restantes se empaquetan si fuera necesario.

- Las rutinas de utilidad cuyos nombres están almacenados en la tabla se mueven desde el MASTER al disco flexible instalado en la unidad 1.

Durante este proceso pudiera suceder que una rutina de utilidad que se encontrase en el disco no esté en el MASTER; de producirse esta situación aparecería el mensaje:

Ø,FD Nombre-de-rutina NOT FOUND

Debe marcarse esta rutina en el listado del directorio y continuarse con el proceso pulsando FA. Aquellas rutinas de utilidad que resultan señaladas por este motivo, deberán transferirse al disco, mediante MOVE, en un proceso independiente.

. Si una rutina de utilidad no pudiera transferirse debido a falta de espacio en el disco de destino aparecerá el mensaje:

1,FD INSUF SPACE ON MEDIA

debe pulsarse FA para continuar el proceso.

. Después de finalizar la actualización, se imprime:

DATE YY/MM/DD REL II.SS.PP VOL.SER.NUMBER *****

y a continuación los nombres de aquellas rutinas de utilidad que no pudieron transferirse por falta de espacio.

. El sistema presenta en pantalla "END OF UPDATE?"

Púlsese FT para actualizar más discos y continúese en el segundo paso o púlsese Y y FA para finalizar.

. Las rutinas de utilidad que no pudieron transferirse debido a que no estaban en el MASTER debe moverse desde el disco de rutinas de utilidad mediante la MOVE.

. Los discos flexibles que no se pudieran actualizar por falta de espacio, deberán reorganizarse utilizando DELETE y MOVE.

RUTINA DE UTILIDAD SELECT

GENERALIDADES

La rutina de utilidad SELECT, proporciona las siguientes posibilidades:

- Copiar registros seleccionados de un archivo existente.
- Cambiar el contenido o tamaño de campos de registros ya existentes.
- Crear nuevos registros que contengan datos seleccionados de registros ya existentes.

Los registros de salida de la rutina SELECT se graban secuencialmente en el archivo de salida, quedando sin modificar los datos del archivo de entrada.

Los archivos de procedencia y destino pueden encontrarse en la misma unidad o en unidades diferentes; para IDPS/LF (I-8150) los archivos deben encontrarse en disco y para IDPS/SF (I-8130) en disco flexible.

Después de asignar los archivos necesarios y cargar la rutina SELECT en memoria, se deben introducir las instrucciones propias de la rutina.

A continuación se resumen las distintas instrucciones:

- Las frases INFILE y OUTFILE describen características de los archivos de entrada y salida.
- Las frases IN y OUT definen campos, a los que se ha de acceder, pertenecientes a los registros de entrada y salida.
- Las frases INCLUDE, IF y MOVE definen los pasos a ejecutar cada vez que SELECT lee un registro del archivo de entrada.
- Las frase END identifica el final del proceso.
- La frase CLEAR anula todas las instrucciones SELECT anteriores.

DESCRIPCIÓN FUNCIONAL

Al ser cargada en memoria, la rutina SELECT acepta instrucciones de dos posibles fuentes:

- Archivo de parámetros:

  Cuando el mismo proceso SELECT ha de utilizarse repetidas veces es conveniente crear un archivo de parámetros mediante el TEXT

EDITOR con las instrucciones apropiadas. Este archivo debe ser asignado al tiempo de ejecutar la rutina SELECT, con lo que las instrucciones serán leídas automáticamente; la instrucción END debe incluirse en último lugar en el archivo de parámetros.

. Entrada por pantalla:

Si no hay asignado un archivo de parámetros, la rutina presenta S? en pantalla, indicando que está dispuesta a aceptar una instrucción; deberán irse introduciendo todas las instrucciones y finalizar con END, con lo que se ejecutan las siguientes operaciones:

1. Lectura de un registro

   SELECT lee registros secuencialmente del archivo de entrada, comenzando con el primero y los almacena en un área de trabajo de entrada. El proceso finaliza al detectarse el fin de archivo durante una operación de lectura.

2. Ejecución del proceso especificado.

   SELECT ejecuta los pasos definidos en las instrucciones INCLUDE, MOVE e IF. Dependiendo de estas instrucciones se ejecutan una o varias de las siguientes operaciones:

   . Permite una operación de grabación (INCLUDE)

   . Mueve un literal o el contenido de un campo a otro campo (MOVE).

   . Accede a un campo del registro de entrada y efectúa una comparación (IF ....)

   . Desautoriza una operación de grabación (IF ... EXCLUDE)

3. Grabación de un registro.

   Si se permitió una operación de grabación o tuvo lugar una transferencia de datos en el paso 2, SELECT graba un registro en el archivo de salida al procesar la instrucción END.

PROCEDIMIENTOS DE OPERACIÓN

Para ejecutar SELECT son necesarias, por una parte instrucciones JCL y por otra las instrucciones propias de la rutina.

INSTRUCCIONES JCL

. Asignación del archivo de entrada

```
ASSI SI TO nombre-físico (n,td)
```

. Asignación del archivo de salida

ASSIGN SO TO nombre-físico (n,td)

. Asignación del archivo de parámetros, esta asignación sólo es necesaria si las instrucciones de SELECT están almacenadas en un archivo de parámetros.

ASSIGN PRM TO nombre-físico (n,td)

. Carga de la rutina en memoria

EXECUTE SELECT (n,td)

INTRUCCIONES DE SELECT

Las instrucciones propias de SELECT pueden clasificarse en tres grupos.

a) Las que definen los archivos de entrada y salida

INFILE

OUTFILE

b) Las que definen campos de los registros de entrada y salida

IN

OUT

c) Las que especifican el proceso a ejecutar

A continuación se describen todas las instrucciones por separado.

INFILE

La instrucción INFILE describe las características del archivo de entrada, es opcional, puesto que si se omite, SELECT toma del directorio las características del archivo que fue asignado con el nombre lógico SI.

SELECT crea un área de trabajo de entrada que ha de contener los registros leídos del archivo de entrada; las dimensiones de la misma se determinan conforme a los datos suministrados en la instrucción INFILE, o en su ausencia los tomados del directorio del disco.

El formato de la instrucción es el siguiente:

INFILE { FIXED / VARIABLE } ,rl,bs

. INFILE identifica la instrucción

. FIXED o VARIABLE indican si los registros del archivo de entrada son de longitud fija o variable respectivamente.

. rl, especifica la longitud del registro; si los registros son de longitud variable debe indicarse la longitud máxima del registro, pero sin incluir los dos caracteres del indicador de longitud variable.

. bs, especifica el tamaño del bloque.

IN

El formato de la instrucción es IN id,fo,fl, $\begin{matrix}9\\x\end{matrix}$,PACKED,SIGNED

. IN identifica la instrucción.

. id, asigna un identificador a un campo determinado, debe introducirse un número cualquiera de caracteres alfanuméricos. Los caracteres válidos son de la A a la Z y del Ø al 9 ; no son válidos los símbolos ni caracteres especiales. El primer carácter del identificador debe ser alfabético. SELECT utiliza únicamente los 8 primeros caracteres si se utilizan más de 8.

. fo, especifica el desplazamiento entre el primer byte del registro (desplazamiento Ø) y el primer byte del campo de que se trate. Si el registro es de longitud variable deben ignorarse los 2 bytes correspondientes al indicador de longitud variable.

. fl, especifica la longitud del campo, bien en bytes, para datos desempaquetados, o en posiciones decimales, para campos empaquetados. Si el dato lleva signo debe aumentarse en un byte o en una posición la longitud para tener en cuenta el signo.

. El resto de los parámetros indican el tipo de campo, si no se indica nada se supone que es alfanumérico.

  . 9 identifica un campo numérico desempaquetado y sin signo

  . x identifica un campo alfanumérico

  . PACKED (OP) identifica un campo numérico empaquetado

  . SIGNED (OS) identifica un campo numérico con signo.

OUTFILE

La instrucción OUTFILE define las características del archivo de salida, es opcional y en caso de omitirse se supone que las características del archivo de entrada y el de salida son las mismas.

SELECT crea un área de trabajo de salida a fin de contener los registros que han de ser grabados en el archivo de salida; las dimensiones de la misma se determinan conforme a los datos suministrados en la instrucción OUTFILE, o en su ausencia, de acuerdo con la longitud del registro de entrada.

El formato de la instrucción es el siguiente:

```
OUTFILE  FIXED
         VARIABLE  ,rl,bs
```

donde los distintos conceptos son idénticos a los explicados en la instrucción INFILE, pero referidos ahora al archivo de salida.

INCLUDE

Debe introducirse una instrucción INCLUDE para autorizar la grabación de un registro en el archivo de salida al procesar la instrucción END.

Se debe utilizar para grabar un registro independientemente de si ha tenido lugar o no movimiento de datos cuando SELECT procesa cada registro de entrada (cuando se detecta la condición especificada en una instrucción IF ... EXCLUDE, SELECT lee automáticamente el siguiente registro de entrada sin grabar nada en el archivo de salida).

El formato de la instrucción INCLUDE es:

```
INCLUDE ALL
```

MOVE

La instrucción MOVE se utiliza para transferir datos a un campo destino.

El dato a transferir puede ser, bien un literal definido en la misma instrucción o el contenido de un campo de un área de trabajo.

El campo de destino puede encontrarse en el área de trabajo de entrada o en el de salida.

SELECT ejecuta la instrucción MOVE sin tener en cuenta las diferencias entre las características de los datos a transferir y el campo de destino, de acuerdo con las siguientes normas:

- SELECT transfiere los datos desde el campo de origen a un campo alfanumérico, comenzando con el caracter situado más a la izquierda.

- SELECT rellena un campo alfanumérico de izquierda a derecha. Si el campo se llena, SELECT trunca el resto de los datos a transferir. Si el dato no llena el campo de destino SELECT rellena el resto del mismo con espacios.

- SELECT transfiere los datos desde el campo origen a un campo numérico comenzando con el carácter situado a la derecha.

. SELECT rellena un campo numérico de derecha a izquierda. Si el campo se llena, SELECT trunca el resto de los datos a transferir. Si el dato no llena el campo de destino, SElect rellena el resto del mismo con ceros.

. No se pueden definir valores numéricos decimales, todos los valores se suponen enteros.

El indicador de final de registro (EOR) es una referencia especial que identifica una posición determina en el registro a grabar en el archivo de salida. Debe especificarse EOR como campo de destino en una instrucción MOVE para almacenar un dato en el registro de salida a partir de la dirección indicada por EOR. Si se transfiere un literal numérico con signo al EOR, SELECT añade un signo más (+) al valor cuando se almacena en el campo de destino. Las funciones de EOR son las siguientes:

. Si el archivo de parámetros contiene instrucciones OUT, el EOR se sitúa con la posición relativa Ø cada vez que SELECT lee un registro. Según se vayan ejecutando pasos del proceso el EOR se va actualizando para indicar en cada momento el final de datos en el área de trabajo de salida.

. Si no existen instrucciones OUT, el EOR se sitúa con la posición del primer byte que sigue al registro de entrada en curso. Si el tamaño del registro de salida (definido en la instrucción OUTFILE) es mayor que el del registro de entrada, debe utilizarse el EOR para almacenar datos en el espacio que sigue al registro de entrada.

El formato de la instrucción MOVE es el siguiente:

```
        lt
        id-1        EOR
MOVE    IRNO    TO  id-2
        IRLEN
```

. MOVE identifica la instrucción

. lt, es un literal o una constante figurativa

. id-1, es un campo definido en una instrucción IN u OUT

. IRNO, es una referencia especial que contiene un contador de registros, es un campo de siete dígitos con signo, su valor es igual a 1 cuando SELECT lee el primer registro, y así sucesivamente.

. IRLEN, es una referencia especial que contiene la longitud del registro en curso como un valor de siete dígitos con signo.

. EOR, es una referencia especial que indica el fin de datos en el área de trabajo de entrada o salida.

. id-2, es un campo definido en una instrucción IN u OUT.

IF

La instrucción IF se utiliza para verificar si se cumple una determinada condición y en su caso, ejecutar una determinada operación LEXCLUDE, INCLUDE o MOVE.

El formato general de la instrucción es:

```
        lt                      lt
IF      id-1     Comparación    id-2     Operación.
        IRNO                    IRNO
        IRLEN                   IRLEN
```

Las comparaciones que se pueden ejecutar son:

- = (Igual a)
- < (menor que)
- > (mayor que)
- =< (igual o menor que)
- => (igual o mayor que)
- <> (menor o mayor que)

pueden establecerse comparaciones complejas mediante las partículas AND y OR; en este caso deben verificarse todas y cada una de las condiciones para que se ejecute la operación.

El siguiente ejemplo ilustra el modo de manejar las condiciones múltiples.

```
IF A B AND (C=D OR E=F) MOVE x TO y.
```

SELECT transferirá el valor de x a y siempre que se verifique que A y B, y simultáneamente al menos una de las condiciones C=D o E=F.

Se pueden definir múltiples operaciones MOVE en una instrucción IF, por ejemplo

```
IF A=B MOVE C TO D MOVE E TO F.
```

Toda instrucción IF debe ir seguida de punto.

Las normas seguidas por la instrucción IF al realizar comparaciones entre campos de características diferentes son las siguientes:

- Si las longitudes de campos alfanuméricos son distintas, se añaden espacios a la derecha del menor de ellos para igualar las longitudes.
- Si las longitudes de campos numéricos son distintas, se añaden ceros a la izquierda del de menor longitud para igualar las longitudes de ambos.

. No pueden considerarse valores decimales, todos los números se suponen enteros.

. Si alguno de los campos tiene signo, se almacena un signo a la derecha de ambos campos, si los signos son iguales se comparan los valores absolutos.

Las operaciones que pueden activarse mediante una instrucción IF son las siguientes:

IF ... EXCLUDE

Si se verifica la condición, automáticamente se pasa a leer el siguiente registro, despreciando el resto de las instrucciones y no grabando el registro en curso.

IF ... INCLUDE

Se debe utilizar esta instrucción para autorizar la grabación cuando SELECT procese la instrucción END.

IF ... MOVE

Si se verifica la condición especificada se efectuarán las tr transferencias que se indiquen, de acuerdo con el ejemplo anterior. Hay que tener en cuenta que la instrucción no puede tener más de 80 caracteres, si fuera necesario habría que incluir varias instrucciones.

END

La instrucción END ha de ser la última introducida, bien manualmente o mediante el archivo de parámetros. Al introducir END, SELECT lee el primer registro del archivo de entrada y ejecuta las operaciones especificadas.

Cada vez que se pasa por la instrucción END durante el proceso, se determina si está autorizada una operación de grabación, como resultado de una instrucción INCLUDE, o si ha habido una transferencia de datos, mediante una operación MOVE. En ambos casos se graba el contenido del área de trabajo que contiene el registro de salida.

El formato de la instrucción es: END

CLEAR

La instrucción CLEAR se utiliza para anular todas las instrucciones SELECT introducidas anteriormente. Cuando la pantalla se borra aparece S? y se acepta la primera nueva instrucción.

RUTINA DE CLASIFICACION PARA DISCO FLEXIBLE (SORT)

GENERALIDADES SOBRE SORT

. La rutina de utilidad SORT proporciona las siguientes posibilidades:

a) Crear un archivo de salida clasificado a partir de uno de entrada no clasificado.

b) Cambiar la organización de un determinado archivo.

. El archivo de entrada debe cumplir los siguientes requisitos:

a) Sus registros deben ser de longitud fija y su tamaño no puede exceder de 512 bytes.

b) Su organización puede ser secuencial o indexada; los archivos relativos no pueden ser procesados por SORT.

c) El archivo debe estar contenido en un solo disco flexible de doble densidad y su tamaño no puede exceder de 1.612 sectores.

. Los registros en el archivo de salida se posicionan en la secuencia determinada por la clave o claves de clasificación, cada una de las cuales es una porción de registro que se utilizará como base de clasificación.

La primera clave introducida es la principal, respecto de la cual se hace una primera clasificación, a igualdad de esta clave los registros serán clasificadas según la clave siguiente y así, sucesivamente.

En el archivo de salida no aparecerán los registros que se encuentran lógicamente eliminados en el de entrada, igualmente se eliminarán los registros con clave duplicada cuando se haya especificado que el archivo es de organización indexada.

Después de ejecutar SORT el archivo de salida se sitúa en la misma posición que el de entrada, quedando éste destruido, por esta razón será obligatorio hacer una copia de seguridad del mismo antes de hacer la clasificación.

ARCHIVO DE TRABAJO

Para la ejecución de SORT son necesarias dos áreas de trabajo, una de ellas es el mismo archivo de entrada, mientras que la otra se obtendrá mediante la creación de un archivo de trabajo, lo que se hará teniendo en cuenta las siguientes consideraciones:

. El archivo de trabajo debe asignarse antes de cargar SORT en memoria, lo cual se hace mediante:

ASSI SWF SWF (unidad,FD),NEW,1612,SC

- El archivo de trabajo se elimina automáticamente después de ejecutar SORT.

- El disco flexible que contiene el archivo de trabajo debe estar inicializado como de densidad doble, de datos y no debe contener ningún archivo de datos.

- El archivo de trabajo debe tener 1.612 sectores asignados al mismo.

CLAVES DE CLASIFICACIÓN

Una clave de clasificación especifica la parte del registro que debe tomarse como base para clasificación, cada una de las cuales puede tener un máximo de 50 bytes.

Hay cinco parámetros a considerar en la clave:

- La posición de la clave es un valor numérico de 1 a 3 bytes, que especifica la posición relativa de la clave a partir del comienzo del registro, entradas válidas son de 0 a 511.
- Longitud de la clave, valor numérico de 1 a 2 bytes, que especifica la longitud de la clave, entradas válidas son de 0 a 50.

- Si la clave de clasificación contiene datos numéricos compactados, debe introducirse una p entre los parámetros de la clave.

- La secuencia de clasificación es normalmente ascendente; no obstante, si hubiera de ser descendente, hay que introducir una D entre los parámetros de la clave.

EJECUCIÓN DE LA CLASIFICACIÓN

La ejecución de la rutina de utilidad SORT puede realizarse de dos formas diferentes:

a) Introduciendo las claves de clasificación por teclado.

b) Creando un archivo de parámetros que contenga las claves de clasificación.

ENTRADA POR TECLADO DE LOS PARÁMETROS

Para ejecutar SORT introduciendo directamente por el teclado los parámetros de las claves de clasificación deben ejecutarse los siguientes pasos:

a) Asignación del archivo de entrada con el nombre lógico de SI.

ASSI SI nombre físico (unidad, FD)

b) Asignación del archivo de trabajo con el nombre lógico de SWF.

ASSI SWF nombre físico (unidad,FD),NE,1612,SC

c) Cargar SORT en memoria mediante:

EXEC SORT (unidad, FD)

d) Una vez cargado en memoria SORT, aparecen en pantalla los distintos mensajes que se contestarán con los parámetros de la clave:

ENTER K FOR KEY, O FOR OUTPUT *

Si se introduce una K se indica que se va a introducir una clave y aparece el siguiente mensaje; si se introduce una o sigue en OUTPUT FILE ORGAN ENTER I OR S, para especificar la organización del archivo de salida.

KEY OFFSET ***

Esta entrada, que ha de estar comprendida entre Ø y 511, indica la posición relativa del comienzo de la clave desde el comienzo del registro.

KEY LENGTH **

Entrada de dos caracteres, comprendida entre 1 y 50, que especifica la longitud en bytes de la clave.

KEY TYPE ENTER S FOR SIGNED *

Si se introduce una s quiere decir que la clave lleva signo; si se introduce una U, o no se hace ninguna entrada antes de pulsar FT, se supone que la clave no tiene signo.

KEY TYPE ENTER P FOR PACKED *

Introdúzcase una P si la clave de clasificación es empaquetada y una U si desempaquetada. Si no se hace ninguna anotación antes de pulsar FT se supone U.

SORT ORDER ENTER D FOR DESCEND

Si se introduce un D en el orden de clasificación será descendente; si se introduce una A o no se hace ninguna entrada antes de pulsar FA, la clasificación se hará ascendente.

Después de realizar esta entrada se vuelve a ENTER K FOR KEY, O FOR OUTPUT, para introducir una nueva clave o dar por terminadas las claves y pasar a

OUTPUT FILE ORGAN ENTER I OR S

Si se introduce una I, el archivo de salida será creado como indexado,

donde la clave del mismo será la clave principal de clasificación, que se considerará no empaquetada, sin signo y ascendente.

Si se introduce una S la organización del archivo de salida será secuencial. Si no se hace ninguna entrada antes de pulsar FT se supondrá que la organización es secuencial.

Una vez se ha pulsado FT empieza a ejecutarse la clasificación.

CLASIFICACIÓN MEDIANTE ARCHIVO DE PARÁMETROS

Cuando la clasificación se realiza mediante un archivo de parámetros, las instrucciones que hay que introducir son las siguientes:

```
ASSI SI nombre-físico (unidad,FD)

ASSI SWF nombre-físico (unidad,FD),NE,1612,SC

ASSI PRM nombre-físico (unidad,FD)

EXECUTE SORT (Unidad,FD)
```

Previamente debe haberse creado un archivo de parámetros que contiene:

a) Caso de archivo secuencial

. Claves de clasificación, que pueden ser 1 a 10, y tendrán el siguiente formato:

```
KE K0,K1,U/S,U/P,A/D*
```

K0 = Posición relativa del primer byte de la clave dentro del registro.

K1 = Longitud de la clave en bytes.

Los demás parámetros tienen el mismo significado que se especificó en la entrada por teclado.

. Marca de fin de archivo de parámetros

```
END
```

b) Caso de archivo indexado

. Clave de clasificación, que tiene el siguiente formato:

```
IN K0,K1,U/S,U/P,A/D
```

donde K0 y K1 tienen el mismo significado que en el caso anterior.

. Marca fin de archivo

END

NOTA: El archivo de parámetros debe estar formateado con registros de longitud fija de 80 bytes y tamaño de bloque de 240 bytes.

RUTINA ESPECIAL DE CLASIFICACION PARA ARCHIVOS INDEXADOS EN DISCO FLEXIBLE (SPECSORT)

GENERALIDADES

La rutina de utilidad SPECSORT se utiliza para clasificar los registros añadidos de un archivo indexado. Esto debiera hacerse siempre que haya un área desclasificada en un archivo indexado antes de ser procesado, pues de lo contrario aumenta sensiblemente el tiempo de acceso, además de no poder accederse al de forma secuencial.

Después de ejecutado SPECSORT el archivo indexado quedará clasificado en orden ascendente; la clave de clasificación se toma del directorio y se considera no empaquetada y sin signo, por lo que no será necesario introducir ningún parámetro.

El archivo de entrada debe estar contenido en un solo disco de doble densidad o cuádruple capacidad.

El archivo de salida (clasificado) se sitúa en la misma posición que el de entrada, por lo que el archivo de origen se destruye, por esta razón es obligatorio hacer una copia de seguridad antes de efectuar la clasificación.

La rutina de utilidad SPECSORT necesita para su ejecución un archivo de trabajo SCRATCH, que debe ser, al menos, de la misma longitud que el de entrada.

EJECUCIÓN DEL SPECSORT

- Asignar el archivo de entrada con nombre lógico SI

  ASSI SI nombre-físico (unidad,FD)

- Crear el archivo de trabajo con nombre lógico SWF

  ASSI SWF nombre-físico (unidad,FD),NE,ns,sc

- Montado en una de las unidades del disco flexible que contiene SPECSORT, cargar la rutina de utilidad en memoria.

  EXEC SPECSORT (Unidad,FD)

- A continuación aparece el mensaje:

  16 mr n,FD XXXXXXXXXX MOUNT

  donde XXXXXXXXXX es el nombre-físico del archivo de entrada.

  La acción a realizar es montar en la unidad n el disco flexible que contiene el archivo de entrada. Pulsar R y FT.

. Aparece el mensaje

Ø8 mr n,FD MOUNT MEDIA

La acción a realizar es montar un disco scratch en la unidad n, pulsar R y FT.

A continuación se ejecuta SPECSORT

RUTINA DE UTILIDAD PARA OBTENER UN RESUMEN DEL LOG DEL SISTEMA

SUMLG

DESCRIPCIÓN FUNCIONAL

La rutina SUMLG copia las anotaciones de los archivos SYSLOG (*) de varios discos flexibles a un nuevo archivo de un disco flexible distinto. Después de ser copiadas estas anotaciones se destruyen.

Esta rutina debe ejecutarse cuando aparece en pantalla un mensaje indicando que el SYSLOG se encuentra cerca del límite de su capacidad o que está lleno.

Después de terminar el programa durante el cual aparecieron los mensajes anteriores, el operador debe procesar el SYSLOG de este disco mediante SUMLG. Los SYSLOG de los demás discos flexibles que componen la aplicación también pueden ser procesados en esta pasada.

La rutina comprende dos fases, una fase de recopilación de los SYSLOG y una segunda fase de clasificación del SUMLG, que pueden ser efectuadas por separado. Cada una de llas requiere un disco flexible que debe contener los overlays proporcionados por OVCOPY.

La clasificación se efectúa utilizando la fecha y número de secuencia como clave, quedando las anotaciones en orden de claves ascendentes. El archivo SUMLOG clasificado puede imprimirse o presentarse en pantalla. Dos o más claves con el mismo valor se sitúan contiguamente en el archivo de salida.

(*) El archivo SYSLOG contiene anotaciones que se producen cada vez que tiene lugar un error de entrada/salida en el sistema.

INSTRUCCIONES PARA EL OPERADOR

Durante la ejecución de esta rutina de utilidad pueden tener lugar situaciones que requieran la acción del operador; en este caso aparecerá un mensaje en pantalla.

La rutina SUMLG deberá ejecutarse atendiendo a los siguientes puntos:

- Efectuéses la rutina de principio de día, si es que no se ha hecho previamente.
- Instáleses un disco flexible que contenga SUMLG.
- Cárguese SUMLG en memoria mediante la siguiente instrucción JCL:

EXECUTE SUMLG (n,FD)

. Después de que haya sido cargada la rutina aparecen en pantalla los siguientes mensajes:

SUMLG UTILITY DPS/SF II SS.PP

DO YOU WANT COLLECT *

. Introdúzcase Y y púlsese FA para comenzar la fase de recopilación de SYSLOG, con lo que aparecerá el mensaje siguiente. (si se anota N la rutina continúa con el mensaje DO YOU WANT TO SORT SUMLOG).

SOURCE DISK UNIT NUMBER *

. Instálese el disco flexible del que han de ser copiadas las anotaciones de SYSLOG. Anótese el número de unidad (0, 1, 2 o 3) de este disco fuente y púlsese FT. Aparecerá a continuación el mensaje:

DESTINATION DISK UNIT NUMBER *

. Instálese un disco flexible que contenga overlays (o bien el archivo SUMLOG generado en una pasada previa de la rutina), anótese el número del disco de destino y púlsese FA.

La rutina crea automáticamente un archivo SUMLOG, y copia las anotaciones de SYSLOG procedentes del disco de origen. Una vez copiadas todas las anotaciones se borra el SYSLOG y aparece en pantalla el siguiente mensaje:

NEXT SOURCE DISK ON UNIT n?

. Instálese el siguiente disco flexible cuyo SYSLOG ha de ser procesado, anótese Y y púlsese FA. Se vuelve a realizar todo el proceso explicado anteriormente, una vez procesados todos los discos flexibles que se desean anótese N y a continuación FA.

Aparecerá el mensaje:

DO YOU WANT TO SORT SUMLOG?

. Anótese Y y púlsese FT con lo que aparecerá el siguiente mensaje. (Si se anota N la rutina continúa con el mensaje END SUMLOG?).

SCRATCH DISK UNIT NUMBER *

. Instálese un disco flexible de datos que contenga únicamente los overlays, anótese el número de unidad y púlsese FA. Este disco flexible se utiliza como archivo de trabajo para efectuar la clasificación del archivo SUMLOG. El archivo de trabajo se borra al finalizar la rutina. A continuación aparece el mensaje:

END SUMLOG?

. Anótese Y para finalizar la rutina y N para comenzar de nuevo.

. El SUMLOG puede presentarse en pantalla, pero deberá imprimirse y limpiarse mediante la siguiente instrucción JCL:

```
DISPLAY SUMLOG (n,FD) TO (Ø,LP),CLEAR
```

. Una vez obtenida la copia impresa del SUMLOG, este puede borrarse utilizando la instrucción:

```
DELETE SUMLOG (n,FD)
```

TIPOS DE DISCOS FLEXIBLES

En el equipo NCR I-8140 tenemos la capacidad de utilizar cuatro tipos distintos de disco flexible:

1. Disco flexible MAESTRO LOCAL DE SOFTWARE (MASTER)
2. Disco flexible de APLICACIONES
3. Disco flexible de DATOS
4. Disco flexible de INTERCAMBIO DE DATOS

DISCO FLEXIBLE MAESTRO LOCAL DE SOFTWARE (VER APÉNDICE A)

Contiene todos los módulos de software del Sistema Operativo del NCR I-8140 y las tablas de traducción al castellano. Los archivos que contiene son los siguientes:

DPSSOLLO

Contiene la parte del Sistema Operativo no residente en memoria (Overlays) y el archivo de errores del sistema.

DPSSFOPS

Contiene la parte del Sistema Operativo residente en la memoria, el intérprete COBOL, la rutina de comienzo de día y la rutina "DEBUG", herramienta que sirve para depurar los programas COBOL.

RORIFILE

Es el archivo utilizado por la rutina de utilidad INQUIRY para salvar el contenido de la memoria mientras se efectúa la consulta de cualquier archivo, recuperándola después a su estado inicial.

LILDIR

Es el Cargador de Nivel 1. Este cargador, almacenado en memoria mediante una función firmware, cargará desde la unidad del sistema, la parte residente en memoria del Sistema Operativo, y la rutina de comienzo de día, a la cual transfiere el control.

MOVE

Es un comando JCL (Job Control Language) para la copia de archivos.

QCOPY

Es un comando del JCL para copiar discos flexibles.

OVCOPY

Es una rutina de utilidad para copiar los overlays del sistema operativo, de un disco a otro.

CHANGE, DISPLAY, DUMP y DELETE

Son comandos del JCL

SPECSORT

Rutina de utilidad especial para clasificar archivos indexados.

SORT

Rutina de utilidad para clasificar archivos

EDIT

Rutina de utilidad para crear hileras de control y para generar y corregir programas fuente COBOL.

FLEXIT

Rutina de utilidad para incializar discos flexibles.

SUMLG

Rutina de utilidad para copiar el archivo log a disco flexible

OBTENCIÓN

El Maestro Local de Software se obtiene directamente del Departamento de Distribución de Software de NCR España, S.A. El procedimiento para obtener una copia del disco es el siguiente:

. Mediante la rutina de utilidad "FLEXIT", se inicializa un disco, especificando que es de Aplicaciones. Es recomendable poner como número de identificación del disco el número de identificación de la versión del software, por ejemplo 030227, al objeto de identificar el software al imprimir el directorio.

. Mediante la rutina de utilidad "QCOPY" obtener una copia imagen del disco flexible MAESTRO LOCAL DE SOFTWARE (MASTER).

DISCO FLEXIBLE DE APLICACIONES

Un disco flexible de Aplicaciones es aquel disco, de doble densidad, con el siguiente contenido:

. Los archivos de software DPSSOLLO y DPSSFOPS

. Cualquier otro tipo de archivo requerido por la aplicación:

- Hileras de Control
- Programas de la Aplicación y sus overlays
- Archivo de Datos
- Rutinas de Utilidad

Un disco flexible de Aplicaciones puede contener hasta 45 archivos.

OBTENCIÓN

. Mediante la rutina de utilidad "FLEXIT", se inicializa un disco especificando que es de Aplicaciones.

  Con esto se crean en el disco flexible los archivos DPSSOLLO y DPSSFOPS, si bien no contiene ningún dato.

. Mediante la rutina de utilidad OVCOPY se copia el contenido de estos archivos de software, desde el disco flexible MAESTRO.

. Con el comando JCL "MOVE" se copian los programas objeto de la aplicación.

. Con el comando JCL "MOVE" se copian desde el MAESTRO, todas las rutinas de utilidad que se estimen necesarias para la aplicación.

. Mediante la rutina de utilidad EDIT, se crean las hileras de control necesarias.

DISCO FLEXIBLE DE DATOS

Un disco flexible de datos es aquel disco de doble densidad con el siguiente contenido:

- El archivo de software DPSSOLLO
- Archivo de datos
- Dependiendo del diseño del sistema, también puede contener rutinas de utilidad y módulos de los programas de aplicación.

Un disco flexible de datos puede contener hasta 45 archivos.

Los discos de trabajo (SCRATCH) utilizados por las rutinas de utilidad (SORT, SPECSORT) son discos flexibles de datos.

OBTENCIÓN

Mediante la rutina de utilidad FLEXIT se inicializa un disco especificando que es de Datos.

. Con esto se crea en el disco flexible el archivo DPSSOLLO, si bien no contiene ningún dato.

. Mediante la rutina de utilidad "OVCOPY" se copia el contenido de este archivo, de software desde el disco flexible MAESTRO.

. Mediante el comando JCL "MOVE", se copian todos los archivos, módulos de programas y rutinas de utilidad que sean necesarias.

DISCO FLEXIBLE DE INTERCAMBIO DE DATOS

Un disco flexible de Intercambio de Datos es un disco de densidad sencilla, que contiene archivos secuenciales.

Este tipo de discos flexibles no contiene archivos de software, puede contener programas de aplicación para hacer intercambios entre distintos sistemas (por ejemplo: NCR I-8200 a NCR I-8140), pero estos programas no pueden ser ejecutados desde este disco, sino que deben ser copiados mediante el comando JCL "MOVE" a un disco flexible de Aplicaciones.

En este tipo de disco flexible los datos están grabados en código EBCDIC, en las impresiones del directorio deben aparecer "STD", debajo de "FILE FORMAT".

Un disco flexible de densidad sencilla puede contener hasta 19 archivos.

OBTENCIÓN

. Mediante la rutina de utilidad FLEXIT se inicializa un disco especificando que es de densidad sencilla.

. Al crear los archivos de intercambio de datos, es necesario especificar I en el comando ASSIGN (NEW) del JCL.

. Ejemplo: ASSIGN A TO B (0,FD), NEW,ns,AP,I

SYSSW 06 .05.58
DISPLAY 05 .10.08

FLEXIBLE DISK DIRECTORY

DATE: 80/10/20 TIME: 00:00:00

DOUBLE DENSITY NUMBER FLAW TRACKS 0 SERIAL NUMBER: 060558 SOFTWARE & DATA FILES

OWNERS ID:

FILENAME	GEN NUM	START	EOSD	EOD	EOF	FILE SIZE	CREATE DATE	RECORD FORMAT	RECORD SIZE	BLOCK SIZE	FILE FORMAT	FILE ORG	FILE TYPE	VOL SEQ	MULTI VOL	KEYS OFF	KEYS LEN
DPSSOLLO	001	0053		0186	0260	0208	80/10/16	FIX	512	512	NCR	SEQ	SOFTW		NONE		
DPSSFOPS	001	0261		0421	0572	0312	80/10/16	FIX	512	512	NCR	SEQ	SOFTW		NONE		
RORIFILE	001	0573		0573	0728	0156	80/10/16	FIX	512	512	NCR	SEQ	SOFTW		NONE		
L1LDIR	001	1873		1907	1906	0034	80/10/16	FIX	512	512	NCR	SEQ	SOFTW		NONE		
QCOPY	001	0729		0795	0794	0066	80/10/20	FIX	512	512	NCR	SEQ	OBJCT		NONE		
OVCOPY	001	0795		0869	0868	0074	80/10/20	FIX	512	512	NCR	SEQ	OBJCT		NONE		
MOVE	001	0869		1011	1010	0142	80/10/20	FIX	512	512	NCR	SEQ	OBJCT		NONE		
DELETE	001	1011		1085	1084	0074	80/10/20	FIX	512	512	NCR	SEQ	OBJCT		NONE		
COBOL	001	1085		1185	1184	0100	80/10/20	FIX	512	512	NCR	SEQ	OBJCT		NONE		
COMPIL	001	1185		1421	1420	0236	80/10/20	FIX	512	512	NCR	SEQ	OBJCT		NONE		
CAREPK	001	1421		1441	1440	0020	80/10/20	FIX	512	512	NCR	SEQ	OBJCT		NONE		
SELECT	001	1441		1553	1552	0112	80/10/20	FIX	512	512	NCR	SEQ	OBJCT		NONE		
DISPLAY	001	1553		1707	1706	0154	80/10/20	FIX	512	512	NCR	SEQ	OBJCT		NONE		
CHANGE	001	1707		1769	1768	0062	80/10/20	FIX	512	512	NCR	SEQ	OBJCT		NONE		
SUMLG	001	1769		1845	1844	0076	80/10/20	FIX	512	512	NCR	SEQ	OBJCT		NONE		

TIPO DE DISCO FLEXIBLE

```
SYSSW     06 .05.58
DISPLAY   05 .10.08                    FLEXIBLE DISK DIRECTORY
                              DATE: 80/10/20                    TIME: 00:00:00
DOUBLE DENSITY          NUMBER FLAW TRACKS  0       SERIAL NUMBER: 060558          SOFTWARE & DATA FILES
                                        OWNERS ID:

            GEN                          FILE  CREATE  RECORD RECORD BLOCK  FILE  FILE  FILE VOL MULTI   KEYS
FILENAME    NUM START EOSD  EOD  EOF SIZE   DATE   FORMAT  SIZE   SIZE FORMAT ORG  TYPE SEQ  VOL  OFF  LEN

DPSSOLLO    001  0053      0190 0260 0208 80/10/16   FIX     512    512    NCR   SEQ SOFTW      NONE
DPSSFOPS    001  0261      0483 0572 0312 80/10/16   FIX     512    512    NCR   SEQ SOFTW      NONE
RORIFILE    001  0573      0573 0728 0156 80/10/16   FIX     512    512    NCR   SEQ SOFTW      NONE
LILDIR      001  1873      1907 1906 0034 80/10/16   FIX     512    512    NCR   SEQ SOFTW      NONE
FLEXIT      001  0729      0791 0790 0062 80/10/16   FIX     512    512    NCR   SEQ OBJCT      NONE
QCOPY       001  0791      0857 0856 0066 80/10/16   FIX     512    512    NCR   SEQ OBJCT      NONE
QVCOPY      001  0857      0931 0930 0074 80/10/16   FIX     512    512    NCR   SEQ OBJCT      NONE
DELETE      001  1115      1189 1188 0074 80/10/20   FIX     512    512    NCR   SEQ OBJCT      NONE
MOVE        001  0973      1115 1114 0142 80/10/16   FIX     512    512    NCR   SEQ OBJCT      NONE
DISPLAY     001  1189      1343 1342 0154 80/10/20   FIX     512    512    NCR   SEQ OBJCT      NONE
CHANGE      001  1343      1405 1404 0062 80/10/20   FIX     512    512    NCR   SEQ OBJCT      NONE
EDIT        001  1405      1469 1468 0064 80/10/20   FIX     512    512    NCR   SEQ OBJCT      NONE
DUMP        001  1469      1573 1572 0104 80/10/20   FIX     512    512    NCR   SEQ OBJCT      NONE
SORT        001  1573      1665 1664 0092 80/10/20   FIX     512    512    NCR   SEQ OBJCT      NONE
SPECSORT    001  1665      1719 1718 0054 80/10/20   FIX     512    512    NCR   SEQ OBJCT      NONE
SUMLG       001  1719      1795 1794 0076 80/10/20   FIX     512    512    NCR   SEQ OBJCT      NONE
```

SYSSW 06 .05.58
DISPLAY 05 .10.08

FLEXIBLE DISK DIRECTORY

DATE: 80/10/20 TIME: 00:00:00

QUAD DENSITY NUMBER FLAW TRACKS 0 SERIAL NUMBER: 060558 SOFTWARE & DATA FILES

OWNERS ID:

	FILENAME	GEN NUM	START	EOSD	EOD	EOF	FILE SIZE	CREATE DATE	RECORD FORMAT	RECORD SIZE	BLOCK SIZE	FILE FORMAT	FILE ORG	FILE TYPE	VOL SEQ	MULTI VOL	KEYS OFF	KEYS LEN
	DPSSOLLO	001	0053		0190	0260	0208	80/10/14	FIX	512	512	NCR	SEQ	SOFTW		NONE		
	DPSSFOPS	001	0261		0483	0572	0312	80/10/14	FIX	512	512	NCR	SEQ	SOFTW		NONE		
	RORIFILE	001	0573		0573	0728	0156	80/10/14	FIX	512	512	NCR	SEQ	SOFTW		NONE		
	L1LDIR	001	1873		1907	1906	0034	80/10/14	FIX	512	512	NCR	SEQ	SOFTW		NONE		
	FLEXIT	001	0729		0791	0790	0062	80/10/14	FIX	512	512	NCR	SEQ	OBJCT		NONE		
	QCOPY	001	0791		0857	0856	0066	80/10/14	FIX	512	512	NCR	SEQ	OBJCT		NONE		
	OVCOPY	001	0857		0931	0930	0074	80/10/14	FIX	512	512	NCR	SEQ	OBJCT		NONE		
DEL	AUTOPAT	001	0931		0973	0972	0042	80/10/14	FIX	512	512	NCR	SEQ	OBJCT		NONE		
	MOVE	001	0973		1115	1114	0142	80/10/14	FIX	512	512	NCR	SEQ	OBJCT		NONE		
	COBOL	001	1115		1215	1214	0100	80/10/14	FIX	512	512	NCR	SEQ	OBJCT		NONE		
	COMPIL	001	1215		1451	1450	0236	80/10/14	FIX	512	512	NCR	SEQ	OBJCT		NONE		
	CAREPK	001	1451		1471	1470	0020	80/10/14	FIX	512	512	NCR	SEQ	OBJCT		NONE		
	DELETE	001	1471		1545	1544	0074	80/10/14	FIX	512	512	NCR	SEQ	OBJCT		NONE		
	CHANGE	001	1545		1607	1606	0062	80/10/14	FIX	512	512	NCR	SEQ	OBJCT		NONE		
	SELECT	001	1607		1719	1718	0112	80/10/14	FIX	512	512	NCR	SEQ	OBJCT		NONE		
	SORT	001	1719		1811	1810	0092	80/10/14	FIX	512	512	NCR	SEQ	OBJCT		NONE		
	SPECSORT	001	1811		1865	1864	0054	80/10/14	FIX	512	512	NCR	SEQ	OBJCT		NONE		
DEL	DATPAT	001	1925		2037	2036	0112	80/10/14	FIX	512	512	NCR	SEQ	OBJCT		NONE		
DEL	DPSLANGS	001	2037		2105	2104	0068	80/10/14	FIX	512	512	NCR	SEQ	OBJCT		NONE		
	DISPLAY	001	2105		2259	2258	0154	80/10/14	FIX	512	512	NCR	SEQ	OBJCT		NONE		
	DUMP	001	2259		2363	2362	0104	80/10/14	FIX	512	512	NCR	SEQ	OBJCT		NONE		
	EDIT	001	2363		2427	2426	0064	80/10/14	FIX	512	512	NCR	SEQ	OBJCT		NONE		
	COV8200	001	2427		2455	2454	0028	80/10/14	FIX	512	512	NCR	SEQ	OBJCT		NONE		
	SUMLG	001	2455		2531	2530	0076	80/10/14	FIX	512	512	NCR	SEQ	OBJCT		NONE		

TIPO DE DISCO FLEXIBLE

ARCHIVOS EN DISCOS FLEXIBLES

En este apartado se analizan los distintos tipos de archivo que se pueden organizar en disco flexible, y está encaminado a informar a los usuarios de NCR I-8140.

Para comprender perfectamente los distintos conceptos que aquí se vierten es conveniente una lectura previa del apartado que trata de los distintos tipos de disco flexible que pueden utilizarse en una instalación de este tipo.

En una segunda parte (Conceptos del Sistema) se establecen ciertas normas acerca de la elección del tipo de organización que debe escogerse para el archivo, según la utilización que vaya a darse al mismo.

Por último se propone una solución al problema de la falta de acceso dinámico en el actual estado de desarrollo del software del NCR I-8140.

TIPOS DE ARCHIVO EN DISCO FLEXIBLE

El sistema NCR I-8140 procesa archivos con organización secuencial, relativa e indexada soportados en discos flexibles de doble densidad (1 0 2 caras) y archivos secuenciales de densidad simple.

DENSIDAD SIMPLE

Los discos flexibles de densidad simple únicamente pueden contener archivos secuenciales con registros de longitud fija y cuyo tamaño no exceda de 128 bytes.

La longitud del bloque es también 128 bytes y el archivo tiene un registro por bloque, independientemente de la longitud del registro.

El objetivo fundamental de este tipo de archivos (archivo de intercambio de datos), es la transferencia de datos entre sistemas NCR y otros sistemas que utilicen discos flexibles, asímismo facilita la transferencia de programas objeto, por ejemplo un programa compilado en NCR I-8200 y que será ejecutado en el NCR I-8140.

DOBLE DENSIDAD

En los discos flexibles de doble densidad (1 o 2 caras) cada bloque abarca 2 sectores y el tamaño de cada sector es de 256 bytes, por tanto cada bloque tiene una longitud de 512 bytes.

Los registros pueden tener una longitud máxima de 512 bytes, si el tamaño del registro es mayor que 256 bytes (y menor que 512, que es el máximo permisible), el bloque contendrá un solo registro; si la longitud del registro es menor que 256 bytes el bloque podrá contener varios registros.

Por ejemplo, si la longitud del registro es 60 bytes el bloque contendrá 8 registros.

Restricciones

. Actualmente sólo se permiten registros de longitud variable en los archivos secuenciales.

. Actualmente se permiten únicamente archivos en un solo volumen.

. Actualmente sólo se permiten los métodos de acceso secuencial y al azar (random).

. Según se vayan distribuyendo las nuevas versiones del software estas restricciones irán desapareciendo.

TIPOS DE ORGANIZACIÓN DE ARCHIVOS

ARCHIVOS SECUENCIALES

Un archivo se dice que tiene organización secuencial cuando todo registro, salvo el primero tiene uno exclusivo que le precede, y todo registro salvo el último, tiene uno exclusivo que le sigue. Los registros se van situando en el archivo en el orden en que van siendo grabados y sólo pueden acceder a ellos de forma secuencial, esto es, sólo pueden ser leídos en el mismo orden en que fueron grabados.

La longitud de los registros puede ser fija o variable.

ARCHIVOS RELATIVOS

En un archivo con organización relativa, a cada registro se le asigna una posición específica dentro del archivo, la clave relativa indica la posición que ocupa el registro físicamente en el archivo, así, el registro con clave relativa 1 estará situado en primer lugar; el que tenga 10 estará en décima posición.

La longitud del registro debe ser fija.

No hay ninguna clave de acceso almacenada en el propio registro.

Los archivos relativos pueden ser accedidos secuencialmente o al azar.

ARCHIVOS INDEXADOS

Un archivo indexado se caracteriza porque cada registro se identifica por el valor de un campo o elemento grupo dentro del registro; este campo o elemento grupo se denomina clave y debe tener la misma longitud y ocupar la misma posición relativa para todos los registros del archivo. Los registros son de longitud fija.

Los archivos indexados pueden ser accedidos en forma secuencial o al azar.

El NCR I-8140 no almacena el índice en el disco flexible, sino que utiliza el método de Indice Residente en Memoria (MRI) (Veáse párrafo Características del Indice Residente en Memoria)

El índice de un archivo indexado proporciona al sistema operativo un método para acceder a los registros, consistente en una tabla en la que se recoge, junto con la clave, la posición física que ocupa el registro.

MÉTODOS DE ACCESO

ACCESO SECUENCIAL

Cualquier tipo de archivo en disco flexible en el NCR I-8140 puede ser accedido de forma secuencial.

- En un archivo con organización secuencial los registros se leen en el mismo orden que fueron grabados.
- En un archivo con organización relativa, los registros se leen en el orden indicado por la clave relativa, empezando por el registro número 1.
- En un archivo con organización indexada los registros se leen en el orden especificado por la clave, empezando con el de menor valor de la misma, siempre que el archivo esté clasificado.

ACCESO AL AZAR

Sólamente puede utilizarse este tipo de acceso con archivos relativos e indexados.

- En archivos relativos, se accede al registro cuya posición relativa dentro del archivo sea la correspondiente al valor contenido en el campo especificado en la frase RELATIVE KEY.
- En los archivos indexados se accede al registro cuya clave sea igual al contenido del campo especificado en la frase RECORD KEY.

ACCESO DINÁMICO

Este tipo de acceso puede utilizarse únicamente con archivos relativos e indexados. Es un combinación de los dos métodos anteriores que permite acceder al mismo archivo tanto al azar como secuencialmente.

CARACTERÍSTICAS DEL ÍNDICE RESIDENTE EN MEMORIA (MRI)

Como ya se dijo anteriormente, en el NCR I-8140 no existe índice en el disco flexible, correspondiente a un archivo indexado, sino que éste se crea en memoria al abrir el archivo con acceso al azar.

El MRI es una tabla que contiene la clave más alta de cada pista, junto con el número de pista que le corresponde, para cada pista clasificada del archivo.

Una vez creado el archivo indexado, y estando éste clasificado, cualquier registro que se añada se incorporará en un área añadida sin clasificar, al final del mismo.

Para acceder a un registro determinado, hay que proporcionar el valor de la clave que le corresponda; el sistema operativo inspecciona el MRI hasta encontrar un valor de la clave igual o inmediatamente superior al dado, toma el número de pista correspondiente e inspecciona secuencialmente la pista hasta encontrar un registro con este valor de clave o termina la pista. En este segundo caso inspecciona el área añadida secuencialmente hasta encontrar el registro buscado o terminar y salir por INVALID KEY.

Es evidente que cuanto mayor sea el área añadida mayor será también el tiempo medio de búsqueda de un registro, por lo que después de haber añadido un cierto número de registros conviene clasificar este área, combinando sus registros con los del resto del archivo, lo que se consigue mediante la rutina de utilidad SPECSORT.

CONCEPTOS DE SISTEMAS

En muchos casos los métodos de organización y acceso a un archivo determinado son evidentes (por ejemplo, un archivo de transacciones creado con datos sin depurar será un archivo secuencial, que se ordenará posteriormente, mediante la rutina de utilidad SORT). Sin embargo, en algunos casos, la complejidad de la aplicación hace que haya que considerar gran número de factores antes de poder diseñar una solución óptima. Esto es particularmente cierto cuando el archivo deba ser utilizado por varios programas distintos.

Por ejemplo, en el caso de acceso al azar puede escogerse organización relativa o indexado. Otra alternativa puede ser crear el archivo como secuencial y posteriormente transformarlo en indexado.

Las siguientes notas intentan auxiliar al analista de sistemas a la hora de escoger el método de procesamiento de archivos y particularmente, en el manejo de archivos indexados.

CASO DE ORGANIZACIÓN INDEXADA. ACCESO AL AZAR

Este tipo de organización/acceso presenta:

a) Caso de Creación (Apertura OUTPUT)

. No es necesario clasificar previamente los documentos de toma de datos para la creación del archivo.

. La cláusula INVALID KEY asegurará que el archivo creado no tenga claves duplicadas.

. Puesto que no existe MRI (ya que el archivo no contenía ningún registro en el momento de abrirlo) por cada registro insertado, se examina desde el principio todo el archivo para determinar si ya existe la clave.

. Con este tipo de apertura el archivo debe crearse con una sola pasada del programa, por lo cual este procedimiento no se puede utilizar para grandes archivos.

b) Caso de Creación (Apertura I-O)

. Este modo de apertura se aplica para programas de mantenimiento donde es preciso efectuar altas, bajas y modificaciones.

. Se recomienda con objeto de disminuir tiempo de proceso, clasificar mediante SPECSORT, con objeto de que queden clasificados todos aquellos registros añadidos.

. El procedimiento, caso de instrucción WRITE, para determinar si existe clave duplicada sería el indicado en "Características del Indice Residente en Memoria".

c) Caso de Archivo de Entrada (Apertura INPUT)

. Este método de organización/apertura se aplicará en procesos de consulta.

. Con objeto de disminuir tiempo de acceso el archivo deberá estar clasificado.

CASO DE ORGANIZACIÓN INDEXADA. ACCESO SECUENCIAL

Este tipo de organización/acceso presenta:

a) Caso de creación (Apertura OUTPUT, entrada por teclado).

. Los documentos base tienen que estar en orden ascendente.

. Este método no se adapta a la creación de archivos de gran tamaño, puesto que debe crearse en una sola pasada.

b) Caso de archivo de entrada (Apertura INPUT)

. Este es un caso típico de impresión de archivos.

. Dado que el acceso es secuencial, el archivo debe estar clasificado.

. Los registros son accedidos sin hacer ninguna referencia de apoyo al MRI.

CASO DE ORGANIZACIÓN SECUENCIAL. ACCESO SECUENCIAL

. Este tipo de organización/acceso se puede aplicar a un archivo que se necesite en el resto de procesos como de organización indexada.

. Se ahorra tiempo en la creación.

. Se pierde la posibilidad de detectar la duplicidad de claves.

. Si se necesita organizarlo indexado habrá que cnvertirlo por medio del SORT, con lo que también se eliminan las claves duplicadas.

. Si se necesitan varias pasadas del programa para conseguir la creación, se puede conseguir fácilmente mediante el OPEN/EXTEND.

ARCHIVO RELATIVO

El archivo relativo, al establecer una correspondencia directa entre el valor de la clave y la posición relativa del registro dentro del archivo, proporciona el método más rápido de acceso directo.

Ejemplo: Si los valores de la clave van de 411001 a 412500, restando 411000 de cada valor de la clave se obtiene que la clave relativa varía entre 1 y 1500.

Sin embargo, si un número no es válido o llega a hacerse absoluto hay que reservar el espacio de un registro, y puede desperdiciarse algo del espacio del archivo.

ACCESO DINÁMICO

El acceso dinámico se implementará en las próximas versiones del software del NCR I-8140. Sin embargo, se describe un método para simular este tipo de acceso, en el apartado DIFERENCIAS COBOL IMOS/COBOL DPS.

DIFERENCIAS ENTRE COBOL IMOS V Y COBOL DPS

INTRODUCCIÓN AL COBOL NCR DPS

El lenguaje de programación COBOL NCR DPS está basado en el COBOL ANSI 1974. La mayoría de los elementos del COBOL DPS son los mismos que para el COBOL IMOS, no obstante existen algunas diferencias que se tratan de explicar en esta publicación.

DIFERENCIAS EN LA IDENTIFICATION DIVISION

Ninguna

DIFERENCIAS EN LA ENVIRONMENT DIVISION

La Environment Division en COBOL DPS tiene la misma estructura que para COBOL IMOS, las únicas diferencias son:

PÁRRAFO SOURCE-COMPUTER

SOURCE-COMPUTER. NCR-DPS (WITH DEBUGGING MODE)

El nombre de ordenador NCR-DPS indica que el equipo en que ha de ser compilado el programa fuente puede ser:

. Un ordenador NCR I-8150

. Un ordenador NCR I-8140

PÁRRAFO OBJECT-COMPUTER

Este párrafo indica el equipo que habrá de ejecutar el programa objeto generado por el compilador.

OBJECT-COMPUTER. NCR-DPS [,MEMORY SIZE entero {WORDS / CHARACTERS / MODULES}]

[,PROGRAM COLLATING SEQUENCE IS nombre-de-alfabeto]

El nombre de ordenador NCR-DPS indica que el equipo en que habrá de ejecutarse el programa puede ser:

. Un ordenador NCR I-8150

. Un ordenador NCR I-8140

La cláusula opcional MEMORY SIZE es puramente informativa, para NCR DPS no tienen sentido los términos WORDS y MODULES, puesto que el equipo trabaja a nivel de carácter, por lo que serán ignoradas por el compilador.

PÁRRAFO FILE-CONTROL

En el párrafo FILE-CONTROL hay varias diferencias entre el COBOL DPS y el COBOL IMOS.

Cláusula ASSIGN

La cláusula ASSIGN determina el tipo de dispositivo hardware asociado con el archivo especificado en el párrafo FILE-CONTROL al que pertenece. Los formatos de la cláusula ASSIGN para las distintas unidades hardware del NCR I-8150 y/o NCR I-8140 son las siguientes:

Unidades Hardware	Cláusula ASSIGN
Disco magnético	ASSIGN TO DISC
Disco magnético flexible	ASSIGN TO DISC
Cassette	ASSIGN TO CASSETTE
Cartucho de datos	ASSIGN TO CASSETTE
Impresora de líneas	ASSIGN TO PRINTER
Manipulador de tarjetas	ASSIGN TO LEDGER
Código de Barras	ASSIGN TO BAR-CODE
Alimentador de papel continuo	ASSIGN TO CFF
Rollo de diario	ASSIGN TO JOURNAL

. El dispositivo hardware denominado impresora de registro visible (VRP), desde el punto de vista del COBOL DPS tiene las siguientes posibilidades:

- El manipulador de fichas, que imprime datos en formatos discontinuos como fichas o tarjetas. Esta posibilidad se referencia mediante el nombre LEDGER.

- El código de barras, que puede leer o grabar un código de barras en una ficha. Se referencia mediante el nombre BAR-CODE.

- El alimentador de papel continuo, que imprime datos en papel continuo y se referencia mediante CFF.

- El diario, que imprime sobre un rollo de diario y se referencia por JOURNAL.

- La VRP puede tener uno o dos manipuladores de tarjetas, denominados manipulador inferior y superior. El inferior debe posicionarse pegado al borde derecho de la impresora, mientras que el superior deberá situarse desplazado a la izquierda respecto del inferior. Si ambos se solapasen, la ficha correspondiente al manipulador inferior deberá insertarse antes que la del superior. Ambos manipuladores se diferencian por el tipo de dispositivo especificado en la instrucción ASSIGN del sistema operativo (JCL), el manipulador inferior debe asignarse como LL, mientras que el superior se asignará como LU.

- Con cada uno de los manipuladores puede ir asociado un código de barras, el dispositivo debe especificarse en la instrucción ASSIGN del sistema operativo como BL para el código de barras asociado con el manipulador inferior y como BU para el asociado con el superior.

- En VRP puede haber uno o dos alimentadores de papel contínuo, diferenciadas por el tipo de dispositivo especificado en la instrucción ASSIGN del JCL, que es C1 para el alimentador de papel contínuo 1 y C2 para el alimentador de papel contínuo 2.

- A fin de permitir la migración hacia otros equipos, el COBOL DPS permite especificar ASSIGN TO PRINTER para archivos correspondientes al alimentador de papel contínuo o al diario. En este caso serán las asignaciones JCL las que definan el dispositivo hardware de que se trata.

Cláusula ACCESS MODE

En COBOL DPS existe una restricción referente a esta cláusula, no puede especificarse ACCESS MODE IS DYNAMIC para un archivo indexado o relativo contenido en disco flexible (solamente 8140). Adviértase que no se producirá ningún mensaje de error en el listado de compilación. En futuras versiones de software se incorporará el acceso dinámico para archivos en disco flexible.

No obstante puede simularse acceso dinámico de la forma siguiente:

Este método juega con la posibilidad de definir en un mismo archivo físico dos archivos lógicos distintos; de esta forma, se podrá utilizar en uno de ellos ACCESS MODE RANDOM y en el otro ACCESS MODE SEQUENTIAL; el primero se utilizará para acceder a un registro cualquiera del archivo y el segundo para leer secuencialmente los registros que se deseen hasta un valor prefijado de la clave. Esta operación puede repetirse tantas veces como sea necesario, no habiendo ninguna restricción en cuanto al registro al que se acceda al RANDOM en las distintas ocasiones.

Habrá que adoptar la precaución de no tener ambos archivos abiertos al mismo tiempo.

Los puntos que debe tener en cuenta el programador para el manejo del archivo son los siguientes:

. Hay que proporcionar dos cláusulas SELECT para dos archivos lógicos distintos, que difieran únicamente en el ACCESS MODE y la RECORD KEY.

```
SELECT NOMBRE 1                SELECT NOMBRE 2

ASSIGN TO DISC                 ASSIGN TO DISC
ORGANIZATION INDEXED           ORGANIZATION INDEXED
ACCESS RANDOM                  ACCESS SEQUENTIAL
```

. Proporcionar un párrafo I-O control con:

La cláusula SAME AREA para ambos archivos.

Cláusula SAME RECORD AREA para ambos archivos.

. La definición de ambos archivos en la FILE SECTION debe ser idéntica, salvo en:

El nombre lógico de ambos.

```
FD NOMBRE 1                    FD NOMBRE 2
```

Sólo será necesario definir el registro en el primero de los archivos, el registro del siguiente archivo se podrá definir con la clave y un FILLER.

. En la cláusula VALUE OF FILE-ID habrá que poner el mismo valor para pasar al sistema operativo un solo nombre lógico.

. La forma de trabajar con este método es:

1. Abrir el archivo NOMBRE 1
2. Acceder al azar al registro deseado, especificando el valor de la clave.
3. Cerrar el archivo NOMBRE 1
4. Abrir el archivo NOMBRE 2
5. Procesar secuencialmente este archivo hasta el registro deseado, desde el que se desee, al cual se accederá mediante una instrucción START, si no ha de ser el primero del archivo.
6. Cerrar el archivo NOMBRE 2; esta serie de aperturas y cierres puede efectuarse tantas veces como sea necesario.

Cláusula OVERLAP

El formato de esta cláusula perteneciente a la FILE-CONTROL es:

OVERLAP

Especifica que la ejecución de una instrucción de entrada/salida, asociada al archivo que se indica, se solapará con la ejecución de las instrucciones que sigan a dicha instrucción:

Las instrucciones de entrada/salida que pueden solaparse para los diferentes tipos de archivos son:

- Secuencial: OPEN, READ, REWRITE, TRANSPORT, WRITE
- Relativo : OPEN, READ, READ NEXT, REWRITE, DELETE, START, WRITE.
- Indexado: OPEN, READ, READ NEXT, REWRITE, DELETE, START, WRITE.

Las instrucciones CLOSE y TEST no pueden solaparse.

La instrucción TEST deberá ejecutarse después de la instrucción de entrada/salida y antes de la siguiente instrucción de este tipo para el mismo archivo, a fin de tener la certeza de que la primera de ellas ha finalizado.

Para los archivos contenidos en cassette no puede utilizarse esta opción.

DIFERENCIAS EN LA DATA DIVISION

En este párrafo incluiremos únicamente el funcionamiento de la cláusula LINAGE de la FILE-SECTION, debido a su importancia para los distintos archivos asociados a la impresora de registro visible.

Cláusula LINAGE

La cláusula LINAGE especifica las dimensiones de una página lógica definida para un archivo, con organización secuencial, asociado con la impresora de línea, alimentador de papel contínuo, manipulador de fichas o rollo diario. Los verbos COBOL DPS que están relacionados con la cláusula LINAGE son WRITE y TRANSPORT.

El formato de la cláusula es:

LINAGE IS { nombre-de-dato-1 / entero-1 } LINES [,WITH FOOTING AT { nombre-de-dato-2 / entero-2 }]

[,LINES AT TOP { nombre-de-dato-3 / entero-3 }] [,LINES AT BOTTOM { nombre-de-dato-4 / entero-4 }]

Las dimensiones de una página lógica vienen definidas por los siguientes elementos:

1. El número de líneas en las cuales puede escribirse y/o espaciarse, denominado cuerpo de página.

2. El número de líneas del margen superior de la página lógica.

3. El número de líneas en el margen inferior de la página lógica.

4. El número de línea, dentro del cuerpo de página, en el que comienza el pie. La primera línea en el cuerpo de página tiene el número 1. El pie termina en la última línea del cuerpo de página.

Frase LINES AT TOP

El entero-3 o el nombre-de-dato-3 de la frase LINES AT TOP especifica el número de líneas del margen superior de la página lógica. No se puede escribir ni espaciar en este margen. El valor de entero-3 o del dato referenciado por nombre-de-dato-3 puede ser cero. El nombre-de-dato-3 debe corresponder a un elemento numérico, entero y sin signo. Si no se incluye la frase LINES AT TOP en la cláusula LINAGE, se supone un margen superior de cero líneas.

Frase LINES AT BOTTOM

El entero-4 o el nombre-de-dato-4 de la frase LINES AT BOTTON especifica el número de líneas en el margen inferior de la página lógica. En este margen no se puede escribir ni espaciar y son aplicables las mismas consideraciones que para el margen superior.

Frase FOOTING

El entero-2 o el nombre-de-dato-2 de la frase FOOTING especifica el número de línea del cuerpo de página en que empieza el pie. La primera línea del cuerpo de página tiene número de línea 1. La última línea del pie coincide con la última línea del cuerpo de página. El nombre-de-dato-2 debe corresponder a un elemento numérico, entero y sin signo. El valor de entero-2 o del dato correspondiente a nombre-de-dato-2 no puede ser mayor que el de entero-1 o nombre-de-dato-1. Esta zona del pie se utiliza para escribir datos tales como totales o subtotales.

Si no se especifica la frase FOOTING, el área de pie de página empieza y termina en la última línea del cuerpo de página.

Registro LINAGE-COUNTER

La presencia de la cláusula LINAGE ocasiona que el compilador COBOL genere un registro especial denominado LINAGE-COUNTER. Este registro es de dos bytes, numérico, empaquetado y sin signo. El valor del LINAGE-COUNTER en cualquier momento representa el número de la línea en curso dentro del cuerpo de página.

El programador puede hacer referencias a este registro mediante instrucciones tales como IF. Sin embargo, no podrá modificar su contenido.

Para cada archivo en cuya descripción se ha incluido una cláusula LINAGE se genera un registro LINAGE-COUNTER. Si existe más de un registro de este tipo, para hacer referencia a los mismos se deberá calificar el nombre. Así, si los archivos SALIDA e INFORME tienen ambos LINAGE, las referencias al registro LINAGE-COUNTER de ambas deberá hacerse de la forma siguiente:

LINAGE-COUNTER IN SALIDA

LINAGE-COUNTER IN INFORME

A continuación se resume la forma en que se modifica el LINAGE-COUNTER:

1. Se sitúa el valor 1 automáticamente al ejecutar una instrucción OPEN OUTPUT.

2. Se pone a 1 automáticamente al ejecutarse una instrucción WRITE o TRANSPORT con la frase ADVANCING PAGE.

3. Se incrementa cada vez que se ejecuta una instrucción WRITE o TRANSPORT con la cláusula ADVANCING identificador/entero. El valor del identificador o del entero determina la cantidad a sumar al LINAGE-COUNTER.

DIFERENCIAS EN LA PROCEDURE DIVISION

En este apartado se recogen aquellos verbos COBOL que pueden tener algún matiz distinto en COBOL DPS respecto de sus correspondientes en COBOL IMOS.

Instrucción READ para archivo de código de barras

En COBOL DPS esta variación de la instrucción READ deja disponible un registro de un archivo asignado a BAR-CODE. El formato general de la instrucción READ para este tipo de archivos es:

READ nombre-de-archivo RECORD [INTO identificador] [;AT END instrucción-imperativa]

El nombre del archivo debe ser igual a uno de los especificados en una cláusula SELECT y en una FD. En la SELECT debe especificarse ASSIGN TO BAR-CODE y la organización y el modo de acceso deben ser SEQUENTIAL.

La presencia de un archivo de código de barras en un programa COBOL requiere la presencia de un archivo asignado al manipulador de fichas correspondiente.

Antes de la ejecución de la instrucción READ, el archivo de código de barras asociado con nombre-de-archivo debe haberse abierto bien INPUT o bien I-O.

El código de barras será leido de una ficha mediante la instrucción READ, y está constituido por los siguientes elementos:

1. Una barra de sincronización que precede a los datos del código de barras.

2. El registro de datos, donde cada byte viene codificado en ocho posiciones; una barra vertical indica un cero binario y un espacio un uno.

3. Una barra de sincronización que sigue a los datos del código de barras.

4. Un carácter de sincronización, consistente en un byte codificado en ocho posiciones binarias.

Si en el momento de intentarse la ejecución de la instrucción READ no hay ninguna ficha insertada se solicita la inserción de ficha, a continuación se lee el código de barras y su contenido se sitúa en el área de entrada correspondiente al archivo especificado en la instrucción READ.

El código de barras puede leerse de una línea distinta de la 1, siempre y cuando haya una instrucción TRANSPORT o WRITE que desplace la ficha antes de la instrucción READ.

Frase AT END

La frase AT END debe especificarse para toda instrucción READ relativa a un archivo de código de barras para el que no se haya programado una instrucción USE en las "declaratives".

Esta condición tiene lugar cuando el registro del código de barras contiene END$ en los cuatro primeros bytes. Cuando se detecta la condición de AT END, tienen lugar las siguientes acciones:

1. Si se especificó un FILE STATUS, se sitúa un valor 10 en el elemento asociado con el file-status.

2. Si se ha especificado la frase AT END en la instrucción READ, se ejecuta la instrucción imperativa correspondiente, no pasando el control a las DECLARATIVES.

3. Si no se ha incluido la frase AT END en la instrucción READ, se ejecutan los procesos de las declaratives, explícita o implícitamente, para este archivo.

Una vez reconocida la condición de fin de archivo, no podrá ejecutarse un READ para el mismo sin haber efectuado previamente un CLOSE con éxito, seguido de un OPEN.

ERROR DE LECTURA PARA EL CÓDIGO DE BARRAS

La situación de error de lectura tiene lugar, para un archivo de código de barras, cuando la línea en que está posicionada la ficha no contiene un código de barras o el carácter de verificación no es válido.

Si no hay código de barras se genera un "file-status" 30 y el

contenido del área reservada para el registro está formado por caracteres NUL, cuya estructura binaria es ØØØØØØØØ. Los caracteres NUL equivalen a la constante figurativa LOW-VALUE en COBOL DPS.

Si el caracter de verificación generado tras la lectura de un código de barras no es igual que el grabado en la ficha se repite la operación, si tras esta operación persiste la desigualdad se genera un file status 3Ø y el contenido del área de registro no se modifica. Las DECLARATIVES para una situación de error en lectura pueden ser utilizadas para permitir la anotación por teclado de información relativa al código de barras cuando este no es válido.

Cuando tiene lugar la situación de error de lectura, se toman las siguientes acciones en el orden que se indica:

1. Si se especificó un FILE-STATUS, se sitúa un valor 30 en el elemento asociado con el mismo.
2. Si se han especificado DECLARATIVES para este archivo, explícita o implícitamente, se transfiere el control a las mismas.
3. Si no se especificaron DECLARATIVES, el control se transfiere a la instrucción siguiente a la READ.

Instrucción TEST

La instrucción TEST de COBOL DPS determina el estado de una operación de entrada/salida para un archivo en el que se ha especificado OVERLAP. El formato de esta instrucción es:

```
TEST nombre-de-archivo  ;ON INCOMPLETE instrucción-imperativa-1
  ;ON ERROR STATUS instrucción-imperativa-1
```

El archivo que se especifica en la instrucción TEST debe tener la cláusula OVERLAP en su FILE-CONTROL.

Antes de la ejecución de la instrucción TEST el archivo debe haber sido abierto INPUT, OUTPUT, I-O o EXTEND.

La ejecución de la instrucción TEST determina si la última operación de entrada/salida ejecutada para ese archivo ha finalizado, la instrucción TEST determina el "status" de la misma a partir del "file-status" en la forma que se indica a continuación:

1. Si el carácter más a la izquierda de "file-status" es igual a cero la última operación de entrada/salida se ha finalizado con éxito y la instrucción TEST transfiere el control a la instrucción que se encuentre inmediatamente a continuación de ésta.

2. Si el carácter más a la izquierda de "file-status" no es igual a cero y se ha especificado la cláusula ON ERROR STATUS en la instrucción TEST, el control se transfiere a la instrucción-imperativa-2. Cualquier DECLARATIVE especificada para este archivo no será ejecutada.

3. Si el carácter más a la izquierda del "file-status" no es igual a cero y no se ha programado la cláusula ON ERROR STATUS se ejecutarán las DECLARATIVES que haya programadas, bien explícita o implícitamente. Si tampoco hubiera declaratives progra-

madas, bien explícita o implícitamente. Si tampoco hubiera DECLARATIVES programadas el control pasaría al sistema operativo.

Si la ejecución de la última operación de entrada/salida para el archivo especificado no ha terminado, la instrucción TEST transfiere el control a la instrucción-imperativa-1. Si no se ha programado la frase ON INCOMPLETE, la instrucción TEST esperará hasta que finalice la operación de entrada/salida.

Instrucción TRANSPORT

La instrucción TRANSPORT es exclusiva del COBOL DPS y su función es posicionar verticalmente un archivo de papel contínuo, de diario o de manipulador de fichas. Su formato es:

```
TRANSPORT nombre-de-archivo ADVANCING { { identificador-1 } [ LINE  ] }
                                      { { entero-1        } [ LINES ] }
                                      {
                                      { nombre mnemotécnico           }
                                      { PAGE                          }
                                      {

[ ; AT { END-OF-PAGE } instrucción imperativa ]
       { EOP         }
```

El archivo referenciado en la instrucción debe tener una de las siguientes cláusulas ASSIGN en la SELECT:

ASSIGN TO CFF

ASSIGN TO JOURNAL

ASSIGN TO LEDGER

su organización y modo de acceso debe ser secuencial.

Antes de poder ejecutar la instrucción TRANSPORT debe haberse efectuado un OPEN OUTPUT sobre este archivo.

La ejecución del TRANSPORT para un archivo de papel contínuo o de rollo diario ocasiona el avance del papel el número de líneas especificado en la frase ADVANCING; para estos archivos no se puede avanzar más de 85 líneas en una sola TRANSPORT. Si el archivo corresponde al manipulador de fichas, en primer lugar se determina si ya hay una ficha insertada en la impresora, si no la hay solicita y, a continuación, la posiciona verticalmente de acuerdo con el número de líneas que se indica en la frase ADVANCING.

Si no se incluye frase ADVANCING, el desplazamiento será de una sola línea. Si se ha especificado LINAGE, el LINAGE-COUNTER se incrementará en el valor especificado en la frase ADVANCING, o en 1 si ésta se omitiera.

Frase ADVANCING nombre-mnemónico

Esta frase ocasiona un avance del papel contínuo, rollo de díario o ficha hasta la línea definida en el párrafo SPACIAL-NAMES de la ENVIRONMENT DIVISION correspondiente al nombre mnemotécnico. Si se utiliza esta opción no debe haber LINAGE para el archivo referenciado.

El valor de n en la cláusula LINE-n del párrafo SPACIAL-NAMES, puede variar entre 1 y 255 y especifica la línea en que debe posicionarse el papel contínuo, rollo de diario o ficha por la instrucción TRANSPORT. Si el valor de n es menor que la línea en curso, en caso de papel contínuo o de diario, el desplazamiento se efectuará hasta la línea correspondiente de la página siguiente; si se trata de manipulador de fichas, la ficha experimentará un desplazamiento negativo hasta la línea indicada.

Frase ADVANCING PAGE

Esta frase ocasiona un desplazamiento de papel hasta la primera línea del cuerpo de la siguiente página lógica, en el caso de papel contínuo o rollo de diario. Para manipulador de fichas ocasiona la expulsión de la ficha. En el caso de los dos primeros dispositivos, tiene que haber programada cláusula LINAGE.

Frase END-OF-PAGE

La frase END-OF-PAGE puede especificarse solamente si se ha programado la cláusula LINAGE para el archivo asociado a la instrucción TRANSPORT que la utilice. Se transfiere el control a la instrucción imperativa cuando se detecta la situación de fin de página.

La condición de fin de página tiene lugar cuando la ejecución de una instrucción TRANSPORT ocasiona un espaciamiento hasta el área correspondiente al pie del cuerpo de página. Esto equivale a que el LINAGE COUNTER iguale o exceda el valor expecificado en la frase FOOTING. Si no se ha especificado FOOTING, esta situación tiene lugar cuando el LINAGE-COUNTER iguale al valor de la última línea del cuerpo de página. Al producirse la condición de salto de página se efectúan las acciones siguientes:

1. El papel contínuo, diario o ficha se mueve conforme a la frase ADVANCING de la instrucción TRANSPORT. El LINAGE- COUNTER se incrementa en el valor especificado en la frase ADVANCING y el valor resultante iguala o excede el comienzo del pie.

2. El control se transfiere a la instrucción imperativa de la frase END-OF-PAGE.

Si la ejecución de una instrucción TRANSPORT ocasiona el que el valor del LINAGE-COUNTER exceda simultáneamente del valor del principio del área de pie de página y del final del cuerpo de página, tiene lugar tanto la condición de fin de página como de salto automático de página. En esta situación las acciones que se ejecutan son las asociadas con el salto automático de página.

Instrucción WRITE para archivo de código de barras

Esta variación de la instrucción WRITE ocasiona la impresión de datos codificados según se explicó en la instrucción READ.

Su formato es:

WRITE nombre-de-registro [FROM identificador]

Si no hay ficha insertada en la impresora, la ejecución de esta instrucción solicita la introducción de una ficha en la impresora, una vez insertada tienen lugar las siguientes acciones:

1. Los datos procedentes del registro especificado se codifican en forma de código de barras.

2. El código de barras se imprime en la línea 1 de la ficha.

3. El código de barras se lee después de ser impreso a fin de efectuar una verificación del mismo.

Una escritura del código de barras puede ser precedida por una instrucción TRANSPORT, a fin de efectuar la impresión en una línea distinta de la primera.

En caso de que la verificación después de la escritura del código de barras no produzca los resultados apetecidos, se generará un "file-status" de valor 30, transfiriéndose el control a las DECLARATIVES, si han sido programadas, o a la siguiente instrucción en caso contrario.

Instrucción WRITE para otros archivos asociados a la VRP

La instrucción WRITE para los archivos asociados a CFF, JOURNAL y LEDGER tiene en COBOL DPS, la misma sintáxis y funciones que la instrucción correspondiente en COBOL IMOS para archivos asociados a PRINTER.

La única restricción es que no es válida la frase

AFTER ADVANCING PAGE

para archivos asociados al manipulador de fichas (LEDGER)